

Technická univerzita v Liberci

FAKULTA PEDAGOGICKÁ

Katedra: Aplikované informatiky

Studijní program: 3. stupeň

Kombinace: Matematika – Informatika

NÁVRH A TVORBA WEBOVÉHO E-LEARNINGOVÉHO PORTÁLU

E-LEARNING PORTAL DESIGN AND DEVELOPMENT

Diplomová práce: 06–FP–KIT–

Autor:

Mgr. Michal VYROUBAL

Podpis:

Adresa:

Pod Kosířem 277 / 29

796 01 Prostějov 1

Vedoucí práce: Mgr. David Kmoch

Konzultant: Mgr. Jiří Vraný

Počet	stran	slov	obrázků	tabulek	pramenů	příloh
	97	20406	85	9	16	3

V Liberci dne 31. 4. 2006.

kopie formuláře zadání diplomové práce

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne 31. dubna 2006

Mgr. Michal Vyroubal

Poděkování

Děkuji všem, kteří přispěli k dokončení mé práce. Vedoucímu diplomové práce Mgr. Davidu Kmochovi děkuji za podnětné rady pro zvýšení úrovně práce a za individuální přístup. Své manželce děkuji za psychickou podporu a toleranci v dobách nejhorších.

NÁVRH A TVORBA WEBOVÉHO E-LEARNINGOVÉHO PORTÁLU

VYROUBAL Michal, Mgr.

DP–2006

Vedoucí DP: Mgr. David Kmoch

Resumé

Diplomová práce stručně uvádí do problematiky e-learningu a internetových technologií. Práce je zaměřena na vytvoření webového e-learningového portálu použitelného v prostředí školy. Podrobně rozebírá návrh databáze portálu a implementaci portálu pomocí PHP. Popisuje pracovní prostředí e-learningového portálu formou manuálu pro učitele. Práce vysvětluje klíčové principy fungování portálu. Vytvořený e-learningový portál je přehledným a efektivním nástrojem učitele pro výuku podporovanou počítačem. E-learningový portál je navržen tak, aby bylo možné jej snadno rozšiřovat.

E-LEARNING PORTAL DESIGN AND DEVELOPMENT

Summary

Thesis briefly introduces to problems of e-learning and technologies of Internet. Thesis is focused on creation of web e-learning portal suitable for environment of school. It describes design of portal's database in detail and it analyses implementation of portal while using PHP. It describes working environment of e-learning portal as a handbook for teacher. Thesis explains key principles of portal's functions. Created e-learning portal is well arranged and effective teacher's tool for teaching supported by computer. The e-learning portal is designed in such way it would be easy to further extend its functions.

Obsah

0 ÚVOD.....	9
1 TEORETICKÝ ÚVOD.....	10
1.1 E-learning.....	10
1.1.1 Důležité pojmy z oblasti e-learningu [2].....	10
1.1.2 Proč začlenit e-learning do výuky.....	13
1.2 Databáze a datová analýza.....	14
1.2.1 Entitně-relační model.....	14
1.2.2 Kardinalita relací.....	15
1.2.3 Konvence při grafickém znázornění ER modelu.....	17
1.3 MySQL.....	20
1.4 WWW, HTML, CSS a validita.....	20
1.4.1 WWW a HTML [10], [11].....	21
1.4.2 CSS.....	21
1.4.3 Validita.....	22
1.5 PHP.....	22
1.5.1 Historie PHP [13].....	23
1.6 Použité nástroje a jejich verze.....	24
2 PORTÁL.....	25
2.1 Funkce a požadavky.....	25
2.1.1 Kurzy.....	25
2.1.2 Uživatelé.....	29
2.1.3 Práva.....	32
2.2 Databáze.....	41
2.3 PHP kód.....	42
2.4 Rozhraní pro moduly.....	43
2.4.1 Informace o umístění portálu a aktuální URL.....	43
2.4.2 Tabulka pro registraci modulu v portálu.....	44
2.4.3 Informace o modulech.....	45
2.4.4 Zvolený modul.....	47
2.4.5 Zvolený kurz.....	47
2.4.6 Uživatelé portálu.....	47
2.4.7 Přístupová práva a role.....	48
3 MODULY.....	51
3.1 Společné vlastnosti modulů.....	51
3.1.1 Ikony pro administraci.....	51
3.1.2 Změna pořadí objektů.....	51
3.1.3 Načasování uveřejnění a stažení.....	53
3.1.4 Připojené obrázky.....	55
3.1.5 Čištění databáze.....	59
3.1.6 Autor objektu.....	59
3.2 Modul Články.....	60
3.2.1 Přístupová práva.....	62
3.2.2 Databáze.....	62

3.3 Modul Soubory.....	63
3.3.1 Přístupová práva.....	65
3.3.2 Databáze.....	65
3.4 Modul Diskuse.....	66
3.4.1 Přístupová práva.....	69
3.4.2 Databáze.....	70
3.5 Modul Testy.....	71
3.5.1 Otázky.....	71
3.5.2 Testy.....	79
3.5.3 Výsledky testů.....	84
3.6 Modul Lekce.....	88
3.6.1 Přístupová práva.....	90
3.6.2 Databáze.....	91
3.6.3 Registrace modulu lekce v portálu.....	92
3.6.4 Registrace nového modulu do modulu Lekce.....	93
3.7 Modul Home.....	94
4 ZÁVĚR.....	97
5 LITERATURA.....	98

Přílohy

PŘÍLOHA 1 – ZDROJOVÝ KÓD NA CD

PŘÍLOHA 2 – MYSQL DATABÁZE

PŘÍLOHA 3 – SCREENSHOTY PORTÁLU

Seznam označení a zkratk

<i>Označení</i>	<i>Význam</i>
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
PHP	Personal Home Page Tools (Personal Home Page Construction Kit)
WWW	World Wide Web
XML	Extensible Markup Language

0 Úvod

Současné školství klade důraz na vytváření mezipředmětových vztahů a používání informačních technologií v hodinách výuky mimo odborný předmět „Informatika a výpočetní technika“, aby se tak studenti naučili pracovat s výpočetní technikou ve spojitosti s nejrůznějšími činnostmi. Proto jsou podporovány projekty, které si kladou za cíl posílit výuku a využití informačních technologií na školách tak, aby studenty počítač provázel celou výukou.

Důležitost uvedeného rozšíření použití výpočetní techniky na školách zdůrazňuje i Ministerstvo školství, mládeže a tělovýchovy, které vede rozsáhlý projekt „Státní informační politika ve vzdělávání“ zahrnující vzdělávání učitelů v používání informačních technologií, výběr vhodných výukových programů a další aktivity. Rovněž Česká školní inspekce při své inspekční činnosti sleduje využití informačních technologií v běžných předmětech.

Tato diplomová práce zapadá do kontextu vývoje současného školství, protože poskytuje učitelům nástroj pro realizaci e-learningu. Nejprve je rozebráno, co je e-learning a naznačeny situace, kdy je možné e-learning používat. Dále je provedena analýza, návrh a realizace e-learningového portálu. Textová část diplomové práce obsahuje návod, s jehož pomocí může učitel zvládnout ovládání portálu. Tento návod je bohatě proložen obrázky ukazujícími rozebíranou vlastnost přímo v prostředí portálu. Navíc je v textové části kompletní rozbor databáze portálu a klíčových míst kódu – dokumentace, která poslouží dalším programátorům, aby mohli na tuto diplomovou práci navázat a rozšiřovat funkčnost portálu. V příloze na CD je kompletní zdrojový kód portálu.

Úvodní část diplomové práce zahrnuje rovněž nástin problematiky několika oblastí informačních technologií, které jsou úzce spojeny s diplomovou prací. Jde o teorii relačních databází a internetové technologie – www, html, css a php. Přílohy obsahují SQL příkazy pro vytvoření databáze e-learningového portálu a pro vložení základních iniciačních záznamů do databáze a několik pohledů na celkové webové rozhraní portálu.

1 Teoretický úvod

Diplomová práce zasahuje svým rozsahem do několika různých oborů od pedagogiky (případně andragogiky) až po webdesign. Nejprve tedy vymezíme základní pojmy těchto oborů v takové míře, v jaké je to nutné vzhledem k hlavní části diplomové práce.

1.1 E-learning

Základním pojmem této práce je e-learning. V podstatě jde o využívání informačních technologií při výuce. Může jít o nejrůznější formy zapojení informačních technologií do výuky od multimediálních výukových kurzů na CD (použitých pro demonstrování učitelem, nebo lépe pro přímou práci studentů), přes Internetové prezentace až po komplexní e-learningové systémy, které poskytují části výukové i testovací, a zahrnují i nejrůznější možnosti interakce a komunikace studentů a učitelů.

V různých předmětech je možnost využití e-learningu různá a je nutné zvážit míru efektivity nasazení e-learningu. Samozřejmě e-learning nemůže nahradit učitele, ale podobně nelze tvrdit, že v daném předmětu není možné využít žádnou formu e-learningu. Otázkou je pouze jaká míra je ještě efektivní a jaká je již samoúčelná.

Nejprve si vymezíme základní pojmy e-learningu, se kterými se může učitel setkat, a poté se zaměříme na konkrétní možnosti, kdy je e-learning výkonným pomocníkem učitele a současně dobrým nástrojem pro studenty.

1.1.1 Důležité pojmy z oblasti e-learningu [2]

1.1.1.1 LMS a LCMS

LMS (Learning Management System)

LMS je systém pro organizaci a řízení výuky a kompetencí. Místo termínu LMS se často používá ekvivalentní termín „řídící systém“.

LCMS (Learning Content Management System)

LCMS je nástroj nebo systém, který slouží pro tvorbu nebo sestavování výukového obsahu.

LMS a LCMS mohou být nezávislé a předávání obsahu vytvořeného pomocí konkrétního LCMS do LMS probíhá na základě dodržování standardů (více viz kapitolu 1.1.1.3 *Standardy e-learningu* na str. 11) při vytváření obsahu. Často jsou však LMS a LCMS integrovány do jediného systému, díky čemuž mohou nabídnout funkce

nad možnosti definovaných standardů (samozřejmě s porušením přenositelnosti do jiných systémů).

E-learningový portál v této práci je systém integrující LMS i LCMS do jediného uživatelského rozhraní, což klade menší nároky na odbornost uživatele pracujícího s portálem, tedy běžného učitele, který zdaleka není aprobevaným informatikem.

1.1.1.2 Blended learning

Blended learning, neboli „smíšená výuka“, je velice často používanou metodou, při které není využíván pouze e-learning, ale i další metody.

Nasazení e-learningu ve školním prostředí zcela jistě bude ve formě smíšené výuky. Půjde o doplnění, rozšíření a obohacení učitelova „běžného“ vyučování – nikoli o jeho nahrazení.

Blended learning kombinuje dvě základní metody výuky – synchronní a asynchronní.

Synchronní výuka

Při synchronní výuce se studenti sejdou ve stejném čase. Místem takového setkání může být učebna, ale může to být i virtuální učebna vytvořená pomocí komunikačních technologií. V obou případech se studenti setkávají, mohou vzájemně komunikovat a reagovat v reálném čase.

Asynchronní výuka

Při asynchronní výuce je učivo předkládáno různým studentům v různém čase. Díky tomu mohou studenti postupovat „svým tempem“ a volit si způsob přijímání informací – například tištěné manuály, knihy a příručky, audio a video, elektronické kurzy atd.

1.1.1.3 Standardy e-learningu

V souvislosti s e-learningem se objevují standardy popisující formát pro výměnu dat mezi LMS a LCMS různých autorů. Tyto standardy a doporučení se týkají jak samotných výukových celků, tak i testů a výsledků testů studentů. Nejznámějšími jsou standardy AICC a SCORM.

Standardy jsou rovněž důležité pro komunikaci LMS a výukového obsahu. LMS musí být schopno spouštět obsah a prezentovat jej studentovi. Při spuštění musí LMS předat výukovému obsahu informace o uživateli, který obsah spouští, informace o datu a času spuštění atp. Výukový obsah naopak předává LMS informace o aktivitách studenta v průběhu kurzu (např. kolik času studoval jednotlivé výukové celky, jakých výsledků dosáhl v testech atp.).

AICC

Standard AICC byl nejrozšířenějším standardem a stále mnoho e-learningových systémů tento standard zahrnuje. Rovněž je k dispozici velké množství výukového obsahu odpovídajícího standardu AICC. V současné době je však více používán standard SCORM.

Více informací o standardu AICC je uvedeno na stránkách

<http://www.aicc.org/>

SCORM

Podle této normy se výukový obsah skládá z učebních objektů – SCO (Shareable Content Object). Takový učební objekt se může dále dělit na další učební objekty (SCO). Učebním objektem je například článek, obrázek, animace, video, ale také struktura obsahující např. několik článků a k nim náležejících multimediálních prvků.

Učební objekt je složen ze samotných výukových dat a z dat popisujících objekt a jeho vnitřní strukturu (meta-data).

Ovládání kurzu a navigace mezi učebními objekty není součástí výukového obsahu, ale je funkcí LMS, který je kompatibilní s normou SCORM. Takový řídicí systém se nazývá SCORM RTE.

Dle této normy je tedy výukový obsah dobře členěn na dílčí celky. Výuková data, popisná data a ovládání kurzu jsou odděleny, díky čemuž je možné s kurzem lépe pracovat, opakovaně využívat výukový obsah a snadno kurz modifikovat.

Pro ověření, zda výukový obsah odpovídá standardu SCORM, je možné použít validátor *SCORM Conformance Test Suit*, který je možné získat na stránkách www.adlnet.org bezplatně.

Více informací o standardu SCORM je uvedeno na stránkách

<http://www.adlnet.org/>

Využití standardů v této práci

Jak již bylo uvedeno, je e-learningový portál, vytvořený v této diplomové práci, integrací LMS a LCMS do jediného systému. Díky tomu je výměna dat mezi LMS a LCMS transparentní a nebylo nutné ji podřizovat standardu SCORM.

Portál je ovšem uspořádán takovým způsobem, aby bylo možné snadno implementovat export, případně import dat odpovídajících standardu SCORM. Učebním objektem (SCO) pak může být například jeden objekt v jednom modulu (jeden článek), nebo všechny objekty modulu, které patří do stejného kurzu (všechny články v kurzu). Učebním objektem by mohl být i celý kurz, který je tvořen objekty z různých modulů; struktura takového SCO je potom patrná z uspořádání definovaného daty modulu *Lekce* (viz kapitolu 3.6 *Modul Lekce* na straně 87).

1.1.2 Proč začlenit e-learning do výuky

Informace, které jsou v e-learningových kurzech uveřejněné, mohou být studentům dostupné stále – při výuce i v době mimo výuku, proto je e-learning výborným doplňkem výuky a pomocníkem učitele.

Na základě e-learningového kurzu mohou být studentům předány souhrnné informace o učivu, které bylo probráno, nebo naopak doplňující informace a úkoly pro talentované žáky. Učitel může využít e-learningový portál pro testování žáků a to za účelem klasifikace, nebo pro poskytnutí informací zpětné vazby studentovi o tom, zda již učivu porozuměl, či jej zvládl v dostatečné míře.

Rovněž může být e-learningový portál použit pro předávání dodatečných podpůrných materiálů pro výuku. Namísto „Tady to si každý do příští hodiny okopíruje,“ přichází pro učitele jednodušší a pro školu méně nákladné „V e-learningovém kurzu jsou materiály, které si na příští hodinu vytisknete a prostudujete.“

E-learningový portál může také sloužit pro zadávání úkolů, na kterých mají studenti pracovat – ať již samostatně v hodině nebo doma. Takto uveřejněné zadání může být jasně a jednoznačně formulováno a studenti se k takovému zadání mohou během zpracovávání úkolu vracet. Samotný proces zadání úkolu je pak pro učitele značně zjednodušen, a čas strávený zadáváním se zkrátí.

Komunikace mezi učiteli a žáky a to ve směru učitel ↔ učitel, učitel ↔ žák a žák ↔ žák může být rovněž zprostředkována e-learningovým portálem, například pomocí diskusí. Přínos takto vedených diskusí je podobný jako při výuce ve třídě. Jde o interakci, jejímž výsledkem je hlubší porozumění učivu. Užitek z ní mají nejen diskutující, ale i „pozorovatelé“ diskuse.

Distanční studium

Vzdělávání se stalo celoživotním procesem, a proto existují na vysokých i středních školách různé formy distančního studia pro doplnění vzdělání a kvalifikace. Taková výuka se samozřejmě neobejde bez přímé výuky, kdy dochází ke kontaktu vyučujícího a studenta. Nicméně čas strávený studentem ve škole je nutné z praktických důvodů optimalizovat a využívat účelně.

E-learning může být dobrou podporou distančního studia. S jeho pomocí může vyučující předávat studentům studijní materiály, zadávat úkoly a umožnit kontrolu procesu autodidakce.

Využitelnost e-learningu v tomto odvětví vzdělávání ukazují mnohé školy a vzdělávací instituty, neboť velice často používají určitou formu e-learningu.

Projektové vyučování

V současné době se ve výuce prosazuje projektové vyučování, které pomáhá vytvářet a posilovat mezipředmětové vztahy. Na projektech mohou spolupracovat studenti

a učitelé z jedné třídy nebo z jedné školy, ale rovněž může jít o studenty a učitele z různých škol, měst, zemí a kontinentů. Pro takovou spolupráci je nutné vybudovat jednoduchý, ale účinný informační systém pro komunikaci a sdílení výsledků práce. Takovým informačním systémem může být právě e-learningový portál.

Každý projekt musí mít výstup v nějaké konkrétní podobě. Těchto podob může být mnoho. Patří k nim i Internetová prezentace. E-learningový portál se tak může stát nástrojem pro prezentování výstupu projektu ve formě článků či galerií, nebo i ve formě e-learningového kurzu (pokud bude taková forma prezentování výstupu projektu zvolena).

Možností využití e-learningového portálu je mnoho. Jednoznačným přínosem z hlediska procesu vzdělávání je připravenost studentů pracovat s daty v elektronické podobě a zkušenost s prací v jednoduchém informačním systému.

1.2 Databáze a datová analýza

Veškerá data e-learningového portálu jsou ukládána do relační databáze. Proto se nyní zaměříme na základy databází a databázových systémů.

Datovou analýzu budeme provádět pomocí entitně relačního modelu (ER modelu). Výsledkem této analýzy je *konceptuální schéma*, které nám pomůže vytvořit si představu o reálné situaci, kterou má databáze popisovat a současně bude sloužit jako podklad při tvorbě databáze. Takto vzniklé schéma je přehledné a snadno srozumitelné. Usnadní tedy nejen implementaci databáze v konkrétním databázovém systému, ale rovněž může posloužit jako základ pro budoucí zdokonalování a rozvíjení databáze.

V následujících podkapitolách provedeme nástin problematiky datové analýzy do takové hloubky, která je nutná pro porozumění hlavní části diplomové práce. Více se může čtenář dozvědět například z [3].

1.2.1 Entitně-relační model

Úvodem této části si nadefinujeme základní pojmy: entita, relace, rekurentní relace, atribut, primární klíč a cizí klíč.

Entita je objekt reálného světa, který je schopen samostatné existence a je jednoznačně rozeznatelný od ostatních objektů. V naší analýze bude tedy entitou například „uživatel Michal Vyroubal, email: vyroubal@fbi.gov, ...“ a další uživatelé jsou dalšími entitami. Pokud všechny entity uživatelů sdružíme, získáme nadřazený pojem **entitní typ**. Dá se tedy říct, že entita je konkrétním výskytem objektu z určitého entitního typu.

Relace je vazba, neboli vztah mezi entitami.

Rekurentní relace vzniká v situaci, kdy jsou ve vztahu entity stejného entitního typu. V e-learningovém portálu se vyskytne hned na dvou místech (kurzy portálu a příspěvky v diskusi). V obou situacích jde o stromovou strukturu, která má být vložena

do databáze. Například v případě entitního typu *KURZ*, je každý kurz (krom kořenového) v relaci se svým nadřazeným kurzem.

Atribut je vlastností entity popřípadě relace mezi entitami. Pro výše zmíněného uživatele je atributem například „Jméno“, „Příjmení“, „Přihlašovací jméno“, „heslo“, ...

Klíčový atribut (primární klíč) je množina atributů (s nenulovou mohutností), jejichž hodnoty jednoznačně identifikují každou entitu. S klíčovými atributy se setkáváme na každém kroku. Jsou to například: pro obyvatele Česka rodné číslo, pro automobily registrační značka (dříve SPZ), pro karoserie vozidel číslo VIN a dokonce i mobilní telefony mají svůj klíčový atribut. Pořídili jste již svému psovi jeho vlastní klíčový atribut?

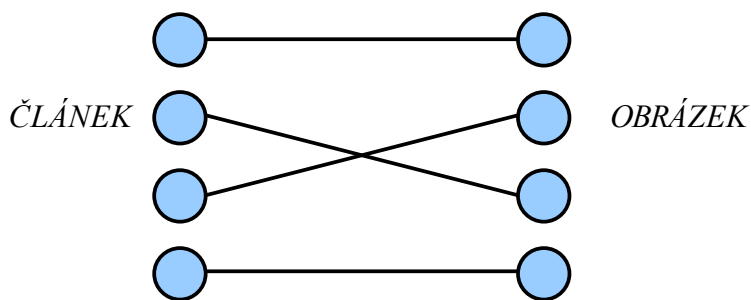
Cizí klíč je atribut, který obsahuje hodnoty klíčového atributu (primárního klíče) jiné entity, neboli jde o odkaz (referenci). Hodnota cizího klíče je shodná s hodnotou klíčového atributu některé z entit entitního typu, na který se cizí klíč odkazuje. Například entitní typ *DOMY VE MĚSTĚ* má jako klíčový atribut *číslo popisné*. Entitní typ *OBYVATELÉ MĚSTA* má jako jeden z atributů i cizí klíč *bydliště*, který jako své hodnoty obsahuje hodnoty klíčového atributu *číslo popisné* příslušné entity *DŮM VE MĚSTĚ*.

1.2.2 Kardinalita relací

Entity vstupují do vzájemných vztahů neboli relací, které mohou být několika typů. Rozdělme si nyní relace podle způsobu, jak se entity na relaci podílejí.

Relace 1:1

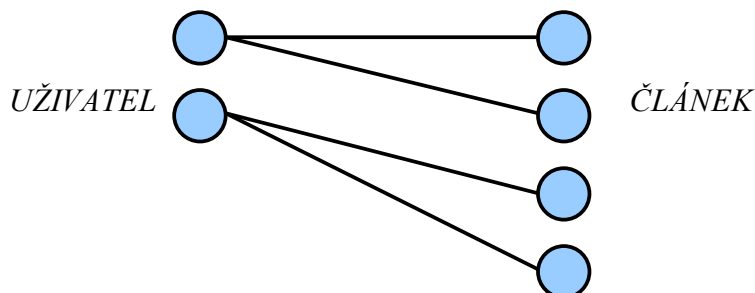
Na obou stranách relace je nejvýše jedna entita. V případě našeho e-learningového portálu je tento typ relace použit například mezi entitními typy *ČLÁNEK* a *OBRÁZEK* a platí, že k danému článku existuje nejvýše jeden obrázek a daný obrázek náleží k právě jednomu článku.



Obrázek 1: Relace 1:1

Relace 1:N

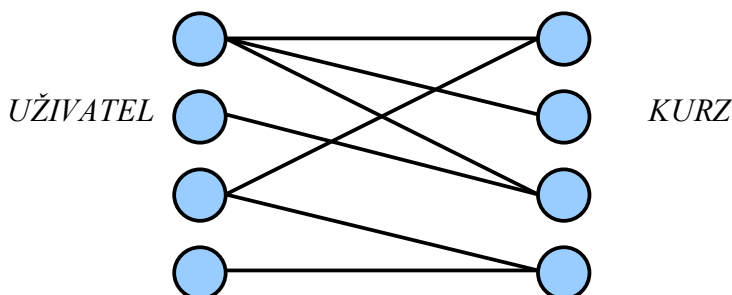
Na jedné straně relace je jedna entita, která může být ve vztahu s více entitami na straně druhé. Relace 1:N je například mezi entitními typy *UŽIVATEL* a *ČLÁNEK*. Jeden uživatel může být autorem více článků, ale pokud vezmeme jeden určitý článek, existuje k němu jednoznačně určený uživatel jakožto autor.



Obrázek 2: Relace 1:N

Relace M:N

Na obou stranách relace může být více entit. V tomto vztahu jsou například entitní typy *UŽIVATEL* a *KURZ*. Jeden uživatel může být redaktorem ve více kurzech a současně jeden kurz může mít více redaktorů.



Obrázek 3: Relace M:N

Relaci M:N není možné v databázi MySQL, která bude použita při tvorbě e-learningového portálu, implementovat přímo, a proto bude provedena **dekompozice relace** M:N na dvě relace typu 1:N. Namísto relace M:N použijeme novou entitu, která se nazývá **průnikový entitní typ**.

1.2.2.1 MIN-MAX IO [7]

Kardinalitu relace je možné vyjádřit pomocí MIN-MAX IO.

Mějme dva entitní typy E_1 a E_2 , které jsou v relaci $R(E_1, E_2)$.

Potom pro $i = 1, 2$ je $E_i(\min_i, \max_i)$,

kde \min_i je minimální počet výskytů entity entitního typu E_i ve vztahové množině R a \max_i je maximální počet výskytů entity entitního typu E_i ve vztahové množině R .

Příklady:

Uvedené příklady odpovídají situacím nastíněným výše.

Relace 1:1 – $R(\text{ČLÁNEK}, \text{OBRÁZEK}), \text{ČLÁNEK}(0, 1), \text{OBRÁZEK}(1, 1)$

Členství entit entitního typu **ČLÁNEK** je v uvedené relaci nepovinné, neboli mohou existovat články, ke kterým není přiřazen žádný obrázek. Naopak entity entitního typu **OBRÁZEK** mají členství povinné, takže každý obrázek musí patřit k nějakému (právě jednomu) článku. (Viz obrázek 1.)

Relace 1:N – $R(\text{UŽIVATEL}, \text{ČLÁNEK}), \text{UŽIVATEL}(0, N), \text{ČLÁNEK}(1, 1)$

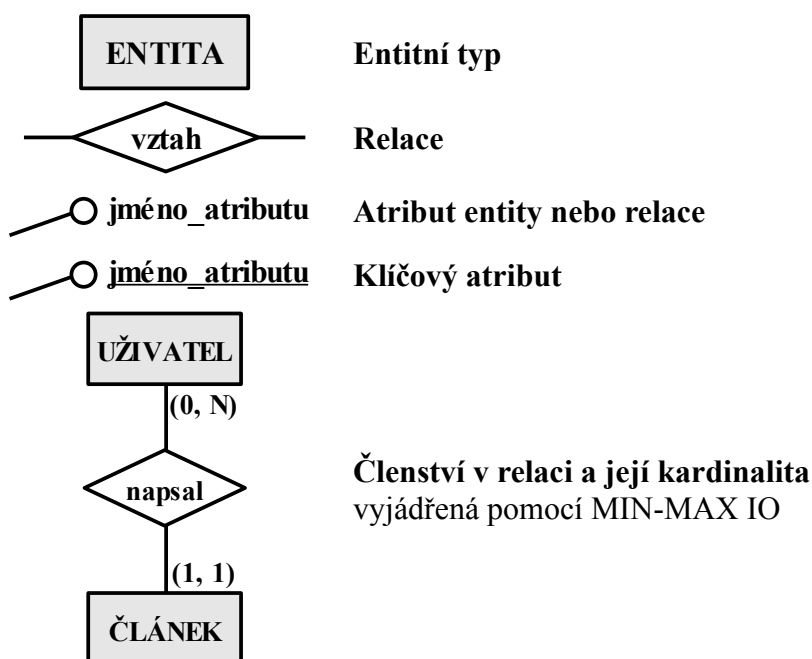
Členství entity **UŽIVATEL** je opět nepovinné – může existovat uživatel, který není autorem žádného článku. (Viz obrázek 2.)

Relace M:N – $R(\text{UŽIVATEL}, \text{KURZ}), \text{UŽIVATEL}(0, M), \text{KURZ}(1, N)$

Členství entity **KURZ** je povinné, protože každý kurz má alespoň jednoho přiřazeného uživatele a to vlastníka kurzu. Nicméně při implementaci portálu přistoupíme k této relaci maličko obecnějším způsobem. (Viz obrázek 3.)

1.2.3 Konvence při grafickém znázornění ER modelu

ER model je možné vyjádřit buď textově, nebo graficky. V této práci se přikloníme ke grafickému vyjádření, protože je velmi přehledné a snadno srozumitelné. Grafické symboly, které budeme používat, jsou shrnuty na obrázku 4. Budeme pracovat se zjednodušenou množinou značek. Například nezohledňujeme typy atributů. Jelikož MySQL databáze pracuje pouze s atomickými (jednoduchými) atributy, nezavádíme značky pro skupinové a vícehodnotové atributy.



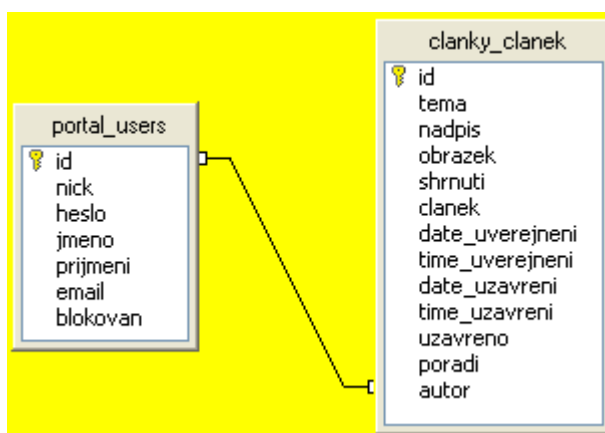
Obrázek 4: Grafické prvky pro znázornění ER modelu

Rady pro zakreslování ER modelu: [3]

- Zakreslovat relace vodorovně, tak aby se daly číst zleva doprava.
- Nepoužívat šikmé čáry.
- Nekřížit čáry.
- Relace typu 1:N zakreslovat s entitou typu N nalevo.
- Kreslit diagram v samostatných celcích, podle logické struktury databáze (celý diagram by byl nepřehledný).
- Pamatovat na to, že výše uvedené body jsou pouze doporučení a rozhodně není možné je splnit vždy a všechny.

Doplňující diagramy

V diagramech, které obsahuje tato práce, nejsou uváděny všechny atributy, ale pouze ty, které jsou pro fungování e-learningového portálu a strukturu databáze důležité, nebo zajímavé. Kompletní seznam atributů a jejich typů je možné najít v příloze 2 ve formě SQL příkazů pro tvorbu tabulek v databázi. Z těchto SQL příkazů je rovněž patrné, jaké datové typy jsou použity pro reprezentaci jednotlivých atributů v databázi.



Obrázek 5: Znázornění tabulek, atributů a relací mezi tabulkami

Pro přehlednost budeme zahrnovat ještě diagram vytvořený v programu OpenOffice.org Base (viz Obrázek 5 – znázorněná situace odpovídá ukázce ER diagramu na obrázku 4), ze kterého je možné zjistit jak názvy všech atributů, tak relace mezi entitami. Rovněž je na něm patrné, které atributy jsou pro entitu klíčové a která pole jsou cizími klíči odkazujícími na klíčové atributy souvisejících entit. Názvy tabulek odpovídají názvům v databázi MySQL (samozřejmě korespondují s názvy entit).

Jsou-li entity v relaci M:N, potom je v tomto diagramu zobrazena dekompozice této relace – je vyznačena tabulka průnikového entitního typu.

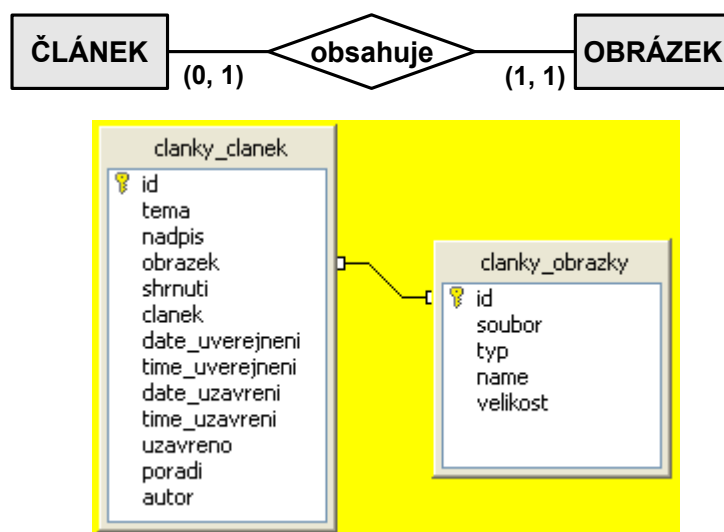
Výhodou ER diagramu je pojmenování relace (například *UŽIVATEL napsal ČLÁNEK*). Budeme tedy uvádět diagramy oba.

Příklady

Nyní zakreslíme pomocí grafického znázornění ER modelu tři příklady, které již byly rozebrány v kapitole 1.2.2 *Kardinalita relací*:

Relace 1:1 – $R(\text{ČLÁNEK}, \text{OBRÁZEK}), \text{ČLÁNEK}(0, 1), \text{OBRÁZEK}(1, 1)$

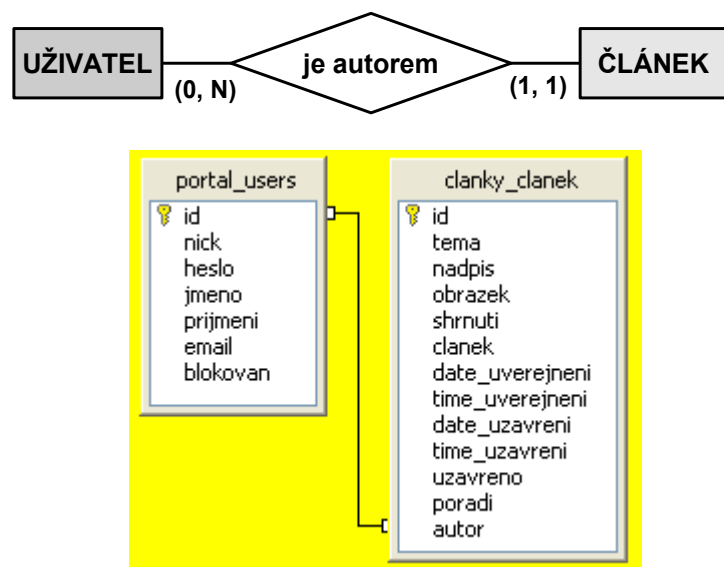
(Viz obrázek 1.)



Obrázek 6: Diagramy relace 1:1

Relace 1:N – $R(\text{UŽIVATEL}, \text{ČLÁNEK}), \text{UŽIVATEL}(0, N), \text{ČLÁNEK}(1, 1)$

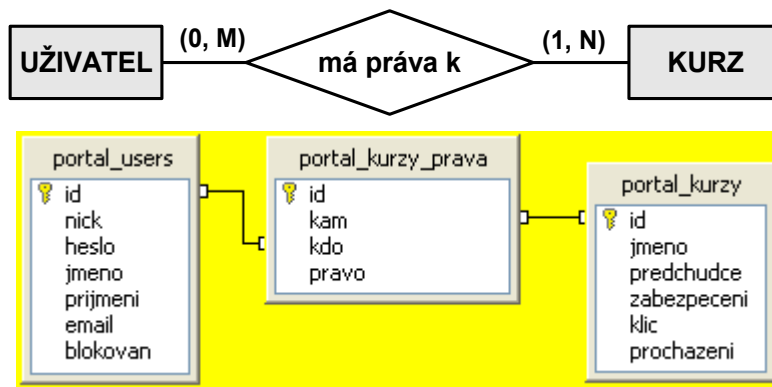
(Viz obrázek 2.)



Obrázek 7: Diagramy relace 1:N

Relace M:N – $R(UŽIVATEL, KURZ), UŽIVATEL(0, M), KURZ(1, N)$

Tabulka *portal_kurzy_prava* vznikla dekompozicí relace. (Viz obrázek 3.)



Obrázek 8: Diagramy relace M:N

1.3 MySQL

Jako databázový systém bylo zvoleno MySQL. MySQL je relační databázový systém, používající dotazovací jazyk SQL pro přístup k datům. Systém MySQL byl vybrán z několika důvodů. Prvním a hlavním důvodem je nekomerční licence, pod kterou je celý systém šířen – konkrétně jde o Open Source. Díky tomu je možné tuto databázi nainstalovat ve školním prostředí bez nutnosti investovat do licencí. Dalším důsledkem nekomerčnosti tohoto databázového systému je jeho rozšířenost na nejrozličnějších webhostingových serverech (od neplacených až po komerční). Proto v zásadě není problém získat webhosting, který nabízí databázi MySQL.

Druhým důvodem je platformní nezávislost, díky které je možné tuto databázi provozovat na Windowsovém i Linuxovém operačním systému (rovněž na systémech Mac OS X, NetWare a UNIX-Like systémech).

Mezi nekomerčními databázemi jsou i jiní adepti, kteří by byli vhodní – například PostgreSQL. Nejsou však tak masově rozšířeni jako MySQL. Samozřejmě celý portál je napsán takovým způsobem, aby bylo možné provést migraci na jiný databázový systém a to pouhým přepsáním metod třídy `databaze` (soubor `class_databaze.php` v adresáři `include`).

1.4 WWW, HTML, CSS a validita

E-learningový portál, který je součástí této diplomové práce, prezentuje své výstupy ve formě webových stránek na Internetu případně školním intranetu. Proto se budeme krátce věnovat základním pojmům z oblasti Internetu spojených se službou WWW. Bude to WWW samotné, jazyk HTML, který je základem pro tvorbu www stránek, a kaskádové styly (CSS), které se používají pro řízení vzhledu www stránek.

1.4.1 WWW a HTML [10], [11]

Nejprve se krátce zastavme u historie WWW, služby sítě Internet, bez jejíž existence by tento e-learningový portál nemohl existovat. Roku 1989 v Ženevě, ve výzkumném centru CERN (Evropská laboratoř pro fyziku částic), začal Tim Berners-Lee vyvíjet distribuovaný hypertextový systém – primárně zamýšlený pro prezentaci dat stávajících informačních systémů. Roku 1990 byla služba WWW v CERNu spuštěna.

Od roku 1991 se tato technologie začala rozšiřovat mimo zdi CERNu, což představovalo vývoj jazyka HTML, sloužícího pro zápis webových stránek, a vývoj klientů (takzvaných prohlížečů) služby WWW.

První specifikace jazyka HTML byla zveřejněna právě v roce 1991. Roku 1993 se objevuje návrh standardu HTML verze 2. Roku 1995 vzniká na základě dokumentu RFC 1866 definitivní podoba standardu HTML verze 2. Roku 1995 je publikován návrh standardu HTML verze 3, který je pak roku 1996 standardizován jako HTML 3.2. V roce 1997 se objevuje HTML verze 4, které je dále revidováno do verze HTML 4.01.

Kombinací HTML verze 4.01 a XML 1.0 vzniká na přelomu let 1999 a 2000 XHTML 1.0. V podstatě jde o přeformulování a rozšíření HTML tak, aby odpovídalo definici jazyka XML. Se stránkami vytvořenými pomocí XHTML je tedy možné pracovat pomocí standardních XML nástrojů.

1.4.2 CSS

CSS se v návrhu poprvé objevuje spolu s definicí HTML verze 3 a standardizováno je roku 1997 – jde o verzi CSS Level 1. Roku 2000 přichází CSS Level 2. V současnosti se pracuje na vývoji verze CSS Level 3. [11]

Kaskádové definice stylů (CSS – Cascading Style Sheet) umožňují oddělit strukturu HTML dokumentu a jeho vzhled. Rovněž je možné pomocí promyšlené definice CSS zajistit jednotný vzhled celého webu a snadno jej modifikovat.

E-learningový portál používá CSS pro možnost jednoduché změny vzhledu portálu bez nutnosti zasahovat do poměrně komplikovaného PHP kódu stránek.

1.4.3 Validita

Významným počinem z hlediska standardizace HTML byl vznik WWW konsorcia, které nadále udržuje a rozvíjí standardy související se službou WWW. Na stránkách tohoto konsorcia je možné získat definice platných standardů, návrhy a diskuse o budoucích standardech, odpovědi na často kladené otázky (FAQ) o WWW a mnoho dalších užitečných informací. Najdeme je na URL

<http://www.w3.org/>

Jak dnes dokazují mnozí tvůrci webových stránek, portálů a služeb, standardy doporučené konsorciem WWW nejsou závazné. Avšak dodržování těchto standardů umožňuje zobrazení stránek v očekávané podobě nezávisle na klientu (prohlížeči) WWW. Stránky, které odpovídají platným standardům, se nazývají validní. Validitu stránek si může kdokoli ověřit na adrese URL

<http://validator.w3.org/>

Validita se sleduje jak u HTML tak i u CSS.

Jako informaci o validitě stránek, může autor na své stránky umístit loga, která jsou rovněž informací o standardu, kterého se autor při tvorbě stránek držel.



Obrázek 9: Informace o validitě stránek

E-learningový portál, který je součástí této diplomové práce generuje stránky v souladu se standardem XHTML 1.0 Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Ačkoli portál sám o sobě generuje validní stránky, je nutné si uvědomit, že validita je rovněž závislá na obsahu vloženém do portálu. Například v modulu *Články* je možné pro vytvoření nového článku použít HTML tagy. Pokud autor článku nebude při jeho tvorbě dodržovat požadavky standardu HTML (respektive XHTML 1.0), nebude stránka portálu ve chvíli zobrazení tohoto článku validní.

Vzhled stránek portálu v této práci je řízen pomocí kaskádových stylů dle definice CSS Level 2.

1.5 PHP

Nástrojem, který byl použit pro tvorbu e-learningového portálu je jazyk PHP. Zaměříme se nyní na PHP samotné – co to je a k čemu to je – a provedeme diskusi, zda je PHP dobrou volbou.

Stránka HTML je uložena na serveru jako soubor a jediné, co je možné s ní udělat, je odeslat ji klientovi. Toto je však velmi nepružné, a proto vznikla celá řada technologií, které umožňují dynamicky vygenerovat HTML stránku v okamžiku, kdy se klient na stránku zeptá.

Dnes velmi rozšířenou technologií pro skriptování na straně serveru je *hypertextový preprocesor PHP*. Základní výhodou PHP je jeho šíření pod nekomerční licencí. Rovněž Apache server, který je možné použít jako webový server při zpracovávání PHP skriptů, je nekomerční produkt. Apache je navíc možné použít jak na platformě Linuxové, tak i Windowsové (na té druhé by to mělo být spíše jen pro účely vývoje nikoli pro ostrý běh

v Internetu či intranetu). Nekomerčnost a platformní nezávislost dvojice PHP + Apache je v ostrém protikladu ke komerčním technologiím ASP + IIS (Active Server Pages + Internet Information Server) firmy Microsoft, vázaným na platformu MS Windows respektive MS Windows Server. Navíc je možná i spolupráce PHP s jinými webovými servery – například se zmiňovaným IIS.

Uvedené skutečnosti determinují PHP jako velmi vhodný nástroj pro tvorbu portálu, který bude nasazen ve školním prostředí, kde rozhodně nejsou prostředky pro zakupování drahých softwarových licencí.

1.5.1 Historie PHP [13]

Jak se PHP objevuje na počítačové scéně? Roku 1994 vytvořil v Perlu Rasmus Lerdorf systém evidence přístupu ke svým webovým stránkám. Později, z důvodu úspory výkonu webového serveru, přepsal tento systém do jazyka C. Systém se postupně stal oblíbeným a na základě připomínek dalších uživatelů se rozšiřoval. Nakonec jej autor uvolňuje spolu s dokumentací pod názvem *Personal Home Page Tools*, později *Personal Home Page Construction Kit*.

Dalším nástrojem vytvořeným panem Lerdorfem byl *Form Interpreter* (FI), který umožňoval přistupovat k databázím pomocí dotazů SQL a uměl zpracovávat formuláře.

Integrací uvedených dvou systémů (PHP a FI) vzniká nástroj PHP/FI 2.0. V podstatě jde o jednoduchý programovací jazyk, který se začleňoval přímo do HTML kódu stránek.

PHP se stále vyvíjí a v současnosti je nejnovější a již běžně používanou verzí PHP 5. Vývoj PHP je velmi progresivní. Objevují se nové funkce a možnosti integrované v PHP, moduly, které rozšiřují možnosti PHP v celé škále používaných webových technologií – multimedia, grafika, databáze atd.

1.6 Použité nástroje a jejich verze

Při vytváření e-learningového portálu byly pro vývoj a testování použity následující systémy a nástroje v uvedených verzích.

Na straně serveru:

- Server 1: **Mandriva Linux release 2006.0** (Official) for i586
 - Kernel 2.6.12-12mdksmp
 - Apache 2.0.54 (Mandriva Linux/PREFORK-13.2.20060mdk)
 - PHP 5.0.4
 - MySQL 4.1.12 + phpMyAdmin 2.6.4-pl1
- Server 2: **Debian 2.1**
 - Kernel 2.4.30
 - Apache 1.3.26
 - PHP 4.3.8
 - MySQL 4.0.23 + phpMyAdmin 2.6.2

Klientský software:

- Mozilla Firefox 1.5.0.1
- Microsoft Internet Explorer 6.0 SP2

2 Portál

Základní a hlavní částí e-learningového portálu je portál samotný, který slouží jako hostitelské prostředí pro jednotlivé moduly. Portál vytváří podmínky pro existenci a fungování modulů a poskytuje jim určité služby (například ověřování uživatelů). Portál sám o sobě tedy umí mnoho, ale pro studenta, který se chce na e-learningovém portálu vzdělávat, by byl portál bez modulů naprosto nezajímavý.

2.1 Funkce a požadavky

E-learning klade na portál určité specifické požadavky, které se týkají zabezpečení a funkčnosti portálu. Pojďme si uvedenou skutečnost nastínit na příkladu.

Učitel může vyžadovat, aby správa v portálu byla distribuována mezi více osob. Například učitel Novák založí kurz, který bude představovat vyučovací předmět, a ostatní učitelé si v tomto kurzu vytvoří „podkurzy“ představující třídy, kde se předmět vyučuje. Jednotliví učitelé by měli mít přístup každý jen do své třídy (svého kurzu) a také by jim mohlo být dovoleno přispět přímo do kurzu předmětu. Naproti tomu učitel Novák by měl mít možnost plně spravovat jak kurz předmětu, tak i kurzy tříd v daném předmětu. Rovněž platí, že uživatel, který je v jednom kurzu studentem, může být v jiném kurzu učitelem.

2.1.1 Kurzy

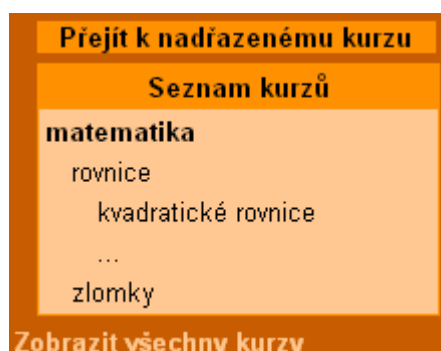
Kurzy portálu jsou uspořádány ve stromové struktuře. Od začátku existuje v portálu kořenový kurz (který nemá žádný nadřazený kurz). Všechny kurzy portálu jsou přímými nebo nepřímými potomky tohoto kořenového kurzu. Kořenový kurz je možné přejmenovat, ale není možné jej smazat (samozřejmě krom přímým zásahem do databáze).

2.1.1.1 Běžná práce s kurzy

V této části se zaměříme na základní operace s kurzy, které v portálu bude, respektive může, provádět kterýkoli uživatel.

Zobrazení seznamu kurzů

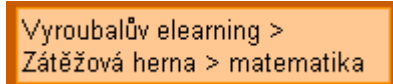
Seznam kurzů je v levé části stránky portálu. Je zobrazen aktuálně zvolený kurz a dvě úrovně podřízených kurzů. Zobrazením dvou podúrovní kurzů se dosahuje rychlejší navigace mezi kurzy.



Obrázek 10: Seznam kurzů

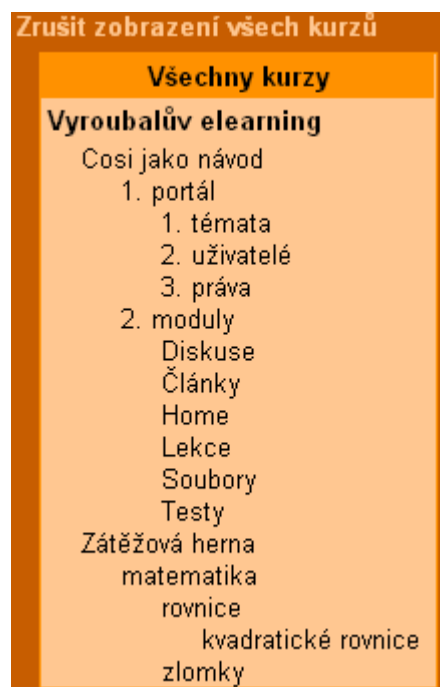
Obrázek 10 ukazuje seznam kurzů, jehož stromová struktura je zakreslena na obrázku 16 na straně 28.

Dále je pro lepší orientaci a rychlejší navigaci uveden seznam předchůdců, který současně slouží jako odkazy pro přechod k nadřazeným kurzům (viz obrázek 11, kde je zvolen kurz *Matematika*).



Obrázek 11: Seznam předchůdců zvoleného kurzu

Dalším zobrazením kurzů je strom kurzů (zobrazuje se odkazem *Zobrazit všechny kurzy*), s jehož pomocí je možné přejít k jakémukoli kurzu portálu přímo. Na obrázku 12 je zobrazení stromu kurzů (jde o zobrazení kurzů od kořenového kurzu portálu).

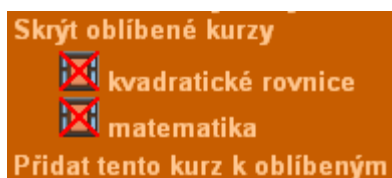


Obrázek 12: Strom kurzů


V určitých situacích může levá část obsahující navigaci mezi kurzy překážet, a proto je možné ji skrýt odkazem *Skrýt seznam kurzů*. Potom se bude zobrazovat pouze seznam předchůdců zvoleného kurzu.

Oblíbené kurzy

Student, který používá e-learningový portál, bude zřejmě navštěvovat pouze několik kurzů. Aby se nemusel vždy komplikovaně „proklikávat“ navigací, umožňuje mu portál spravovat oblíbené kurzy. Seznam oblíbených kurzů je součástí databáze portálu, takže je uživateli přístupný odkudkoli. Pro využití této funkce portálu musí být uživatel registrovaný a přihlášen.



Obrázek 13: Správa oblíbených kurzů

U každého „oblíbeného kurzu“ je umístěna ikona  pro jeho odstranění z oblíbených.

Kurzy, ke kterým má uživatel právo

Jde opět o zrychlení navigace mezi kurzy. Uživatel se může snadno přepnout do kurzů, ve kterých figuruje ať již jako učitel nebo jako student. Jednotlivé role, které mohou být uživateli v kurzu přiřazeny, budou rozebrány v dalších kapitolách.

Skrýt kurzy, ke kterým mám právo			
Vlastníte:	Jste redaktorem:	Jste přispěvatelem:	Jste čtenářem:
zlomky	rovnice	Nejste přispěvatelem žádného kurzu	Zátěžová herna

Obrázek 14: Kurzy, ke kterým má uživatel právo

2.1.1.2 Administrace kurzů

Veškerá správa kurzů se odvíjí od aktuálně zvoleného kurzu. Uživatel se nejprve přihlásí, poté vstoupí do kurzu a jsou mu přidělena oprávnění, na jejichž základě může nebo nemůže provádět administraci kurzů (jaké uživatelské role jsou potřebné pro jednotlivé úkony v portálu, ukazuje kapitola 2.1.3.2 *Práva uživatelů při práci s kurzy* na straně 35).

Nový kurz

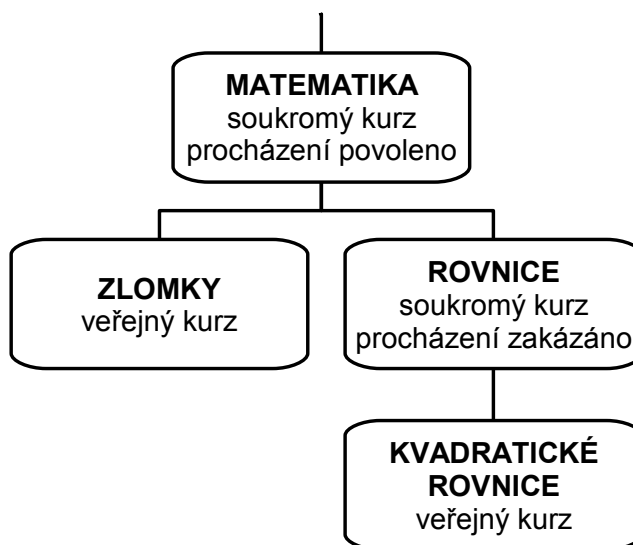
Odkazem *Přidat sem kurz* je možné vytvořit „podkurz“ v aktuálně zvoleném kurzu. Na obrázku 15 je formulář, který slouží pro vytvoření kurzu.

Název kurzu:	<input type="text"/>
Úroveň zabezpečení kurzu:	veřejné <input type="button" value="v"/>
Pro soukromý kurz zadejte vstupní klíč:	<input type="text"/>
Povolit procházení kurzem:	povolit <input type="button" value="v"/>
<input type="button" value="Odeslat"/>	

Obrázek 15: Formulář pro vytvoření kurzu

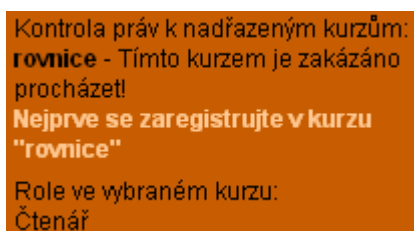
Při vytváření kurzu je možné zadat *úroveň zabezpečení kurzu*. Úroveň zabezpečení nabývá dvou hodnot: *veřejný* a *soukromý*. Veřejný kurz je přístupný všem uživatelům (tedy i anonymním neregistrovaným návštěvníkům portálu). Soukromý kurz je přístupný pouze přihlášeným uživatelům, kteří jsou při prvním vstupu do kurzu dotázáni na *vstupní klíč*, na základě kterého jim je povolen nebo odepřen přístup. Pokud jednou proběhne ověření uživatele na základě klíče, nebude již při dalších přístupech na *klíč* znovu dotazován. (Viz též *Obrázek 27: Ověření vstupního klíče* na straně 33 a rozbor uživatelských rolí uvedený v celé kapitole 2.1.3.1.)

Vzhledem k tomu, že jsou kurzy uspořádány ve stromu, vzniká otázka, jak se má portál chovat u kurzů, které jsou potomky (ať již přímými nebo nepřímými) soukromého kurzu. Má povolit vstup do těchto podřízených kurzů nebo ne? Toto chování je možné nastavit ve vlastnosti *povolit procházení kurzem*. Pokud je procházení kurzem povoleno, může uživatel získat přístup k „podkurzu“ v soukromém kurzu, aniž by nejprve musel získat přístup k nadřazenému soukromému kurzu. Naproti tomu zakázané procházení kurzem způsobí, že uživatel, který chce vstoupit do jakéhokoli „podkurzu“ v soukromém kurzu, musí nejprve získat přístup k tomuto nadřazenému soukromému kurzu – buď je dotázán na klíč, nebo pokud již v minulosti správný klíč zadal, může vstoupit.



Obrázek 16: Procházení soukromými kurzy

Pojďme si celou situaci ukázat na příkladu: Uživatel může volně vstoupit do kurzu *Zlomky* (viz Obrázek 16 – kurz *Matematika* nemusí být kořenovým kurzem, ale budeme předpokládat, že všechny jemu nadřazené kurzy jsou veřejné). Při vstupu do kurzů *Matematika* nebo *Rovnice* bude uživatel dotázán na vstupní klíče těchto kurzů. Při pokusu o vstup do kurzu *Kvadratické rovnice* bude nejprve ověřen přístup ke kurzu *Rovnice* (situace na obrázku 17) – uživatel musí zadat klíč ke kurzu *Rovnice* (pokud tak již dříve neučinil) – a na základě tohoto ověření mu bude povolen nebo zamítnut přístup ke zvolenému kurzu *Kvadratické rovnice*.



Obrázek 17: Zákaz procházení kurzem

Úprava kurzu

Pomocí odkazu *Upravit tento kurz* je možné změnit všechny údaje, které byly zadány při vytváření kurzu. Tímto způsobem je možné přejmenovat kořenový kurz. Dále je možné změnit jeho *úroveň zabezpečení* a povolit mu nebo zakázat *procházení*, čímž se v podstatě nastavuje zabezpečení celého portálu. Pokud bude mít kořenový kurz *úroveň zabezpečení* soukromý a bude mít zakázané *procházení*, budou všechny kurzy přístupné až po zadání vstupního klíče v kořenovém kurzu.

Výmaz kurzu

Hypertextový odkaz *Smazat tento kurz* slouží k odstranění aktuálně zvoleného kurzu spolu s celým podstromem kurzů, který je mu podřízen. Pokud je tedy v situaci z obrázku 16 aktuálně zvoleným kurzem *Matematika*, smažeme kurz *Matematika* a spolu s ním budou odstraněny i kurzy *Zlomky*, *Rovnice* a *Kvadratické rovnice*.

Výjimku při mazání kurzů tvoří kořenový kurz, který není možné uvedeným postupem odstranit.

Hlavní (kořenový) kurz

Kořenový kurz je v portálu od samého začátku. Týkají se jej jistá specifika, z nichž mnohá již byla uvedena výše. Poslední zvláštností je, že kořenový kurz neobsahuje v modulech žádné objekty, protože pokud je zvolen kořenový kurz, neprovádí se include hlavního souboru modulu.

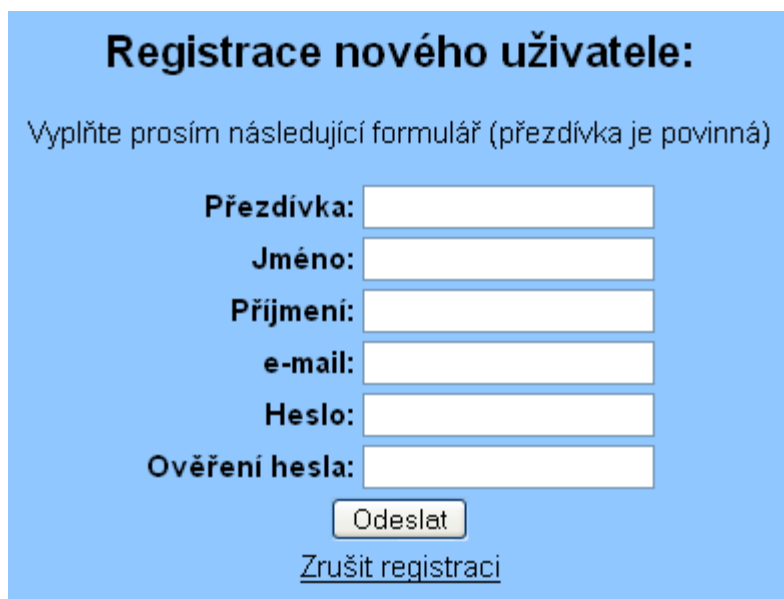
2.1.2 Uživatelé

Studenti a učitelé, kteří přistupují k portálu, potřebují pro efektivní práci s portálem přístup na základě uživatelského jména (přezdívký) a hesla. Jde o základní vlastnost všech víceuživatelských systémů. Na základě provedené autentifikace je potom uživateli umožněno s portálem pracovat. Nyní se zaměříme na práci v portálu, která se týká jakéhokoli uživatele.

Registrace uživatele

Uživatel se může do e-learningového portálu zaregistrovat sám pomocí registračního formuláře. V tomto formuláři jsou dvě omezení. Uživatelské jméno

(přezdívkou) musí být vyplněno a heslo je zadáváno dvakrát shodně (běžná obrana proti překlepům).



Registrace nového uživatele:

Vyplňte prosím následující formulář (přezdívkou je povinná)

Přezdívkou:

Jméno:

Příjmení:

e-mail:

Heslo:

Ověření hesla:

[Zrušit registraci](#)

Obrázek 18: Registrace nového uživatele

Druhou možností jak registrovat nového uživatele je vytvoření uživatele administrátorem portálu. O této možnosti bude pojednáno dále.

Přihlášení uživatele

Při přihlašování uživatele se ověřuje heslo uživatele proti databázi. V databázi není heslo uloženo v otevřeném tvaru, ale je uchováván hash hesla, vytvořený pomocí *jednosměrné hashovací funkce MD5*.



Přihlášení k portálu:

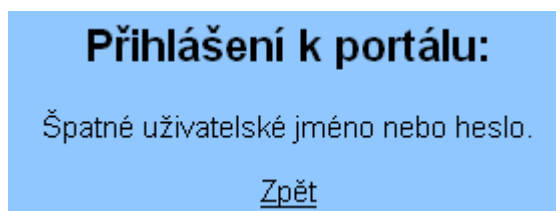
Přezdívkou:

Heslo:

[Zpět](#)

Obrázek 19: Formulář pro přihlášení uživatele

Pokud je zadáno špatné uživatelské jméno nebo heslo, je (v obou případech) vygenerováno jednotné chybové hlášení. Jde o běžný postup, jak mást případného „zkoušeče“ uživatelů a hesel.



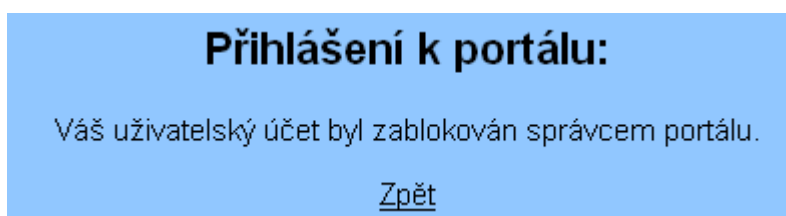
Přihlášení k portálu:

Špatné uživatelské jméno nebo heslo.

[Zpět](#)

Obrázek 20: Chybně zadané přihlašovací údaje

Uživatel může být zablokován administrátorem portálu. Po dobu, kdy je uživatel blokován, mu nebude umožněno přihlášení k portálu.



Přihlášení k portálu:

Váš uživatelský účet byl zablokován správcem portálu.

[Zpět](#)

Obrázek 21: Pokus o přihlášení blokováného uživatele

Změna registračních údajů

Údaje zadané při registraci může uživatel (pokud je přihlášen) změnit. Uživatelské jméno (přezdívkou) není možné změnit. Technicky by to samozřejmě proveditelné bylo, ale není žádoucí, aby uživatel tímto způsobem změnil identitu.

Pole pro změnu hesla jsou nevyplněná. Pokud do nich uživatel nic nenapiše, zůstane původní heslo. Pro obranu proti překlepům je i zde zadáváno heslo dvakrát.



Změna registračních údajů:

Přezdívká: vyroubal

Jméno:

Příjmení:

e-mail:

Heslo:

Ověření hesla:

[Zpět](#)

Obrázek 22: Formulář pro změnu registračních údajů

Změnit údaje zadané uživatelem při registraci má rovněž možnost administrátor portálu; uživatelské jméno (přezdívkou) uživatele však změnit nemůže. Tato funkce portálu bude rozebrána dále.

2.1.3 Práva

Pro řízení přístupu k jednotlivým funkcím portálu je využito ověřování uživatelů na úrovni kurzů. Pro přehlednění celé situace je v portálu definováno několik rolí a uživatel, který k portálu přistupuje, je jedna z těchto rolí přidělena. K této roli se váže sada oprávnění v portálu a v jednotlivých modulech. Je tedy možné říct, že role (respektive k nim náležející práva) umožňují nebo naopak zakazují vykonávat určité úkony – od vstupu do kurzu až po administraci kurzů a uživatelů.

V této části se zaměříme na role jako takové a k nim náležející oprávnění v portálu. Oprávnění, která jsou navázána k rolím v modulech, budou shrnuta vždy v kapitole věnované tomu kterému modulu.

2.1.3.1 Uživatelské role

Vzhledem k tomu, že přidělování rolí uživatelům je prováděno při vstupu do konkrétního kurzu (tedy na úrovni kurzů), může mít jeden uživatel v různých kurzech různé role.

Název role, která byla uživateli přidělena, se zobrazuje na stránce portálu v části nad seznamem kurzů.

Nyní si v obecné rovině ukážeme, jakých rolí může uživatel nabývat a jaké postavení v portálu jednotlivé role zaujímají.

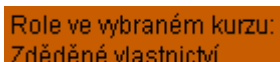
Vlastník

Uživatel v roli vlastníka má v kurzu nejvyšší oprávnění. Stejně jako u dalších rolí, může být v kurzu více uživatelů s rolí *vlastník*. Úkolem a zodpovědností vlastníka je kompletní péče o kurz a všechny jemu podřízené kurzy – respektive o celý podstrom kurzů podřízených vlastněnému kurzu. Proto uživatel dědí vlastnické právo do přímo i nepřímo podřízených kurzů.

Vlastník ručí za obsah jednotlivých modulů i za přidělování a odebírání rolí dalším uživatelům. Proto by měly moduly respektovat pozici vlastníka a přidělovat mu maximální možná privilegia.



Obrázek 23: Vlastník kurzu



Obrázek 24: Vlastnické právo zděděné z nadřazeného kurzu

Redaktor

Role redaktora rovněž spočívá v péči o obsah kurzu, a proto by mu měly moduly poskytnout dostatečné pravomoci. Rozdíl mezi vlastníkem a redaktorem je v tom, že role

redaktora je omezena pouze na jeden kurz (nepřenáší se na podřízené kurzy). Redaktor je ovšem schopen založit si vlastní „podkurz“ a stát se jeho vlastníkem. Tato možnost odpovídá myšlence distribuce a decentralizace pravomocí v portálu.

Redaktor má určitou míru pravomocí nad rolemi uživatelů v kurzu. Samozřejmě nemůže spravovat uživatele s rolí *vlastník*, aby nemohl (mimo jiné) sám sebe povýšit na vlastníka.

Role ve vybraném kurzu:
Redaktor

Obrázek 25: Redaktor kurzu

Příspěvatel

Role příspěvatele umožňuje mít v portálu uživatele, kteří mohou (jak název role napovídá) přispět do jednotlivých modulů takovým způsobem, jak jim to moduly umožní, ale není jim dovoleno zasahovat do příspěvků ostatních uživatelů, což je velký rozdíl v porovnání s vlastníkem a redaktorem. Své vlastní příspěvky samozřejmě může příspěvatel spravovat v plné míře.

Příspěvatel rovněž nemá žádná oprávnění ke správě rolí jiných uživatelů.

Role ve vybraném kurzu:
Příspěvatel

Obrázek 26: Příspěvatel kurzu

Čtenář přihlášený

Přihlášený uživatel, kterému byl umožněn vstup do kurzu, získává automaticky roli *čtenář přihlášený*.

U soukromých kurzů (viz kapitola 2.1.1.2 na str. 28 – *úroveň zabezpečení kurzu*) je pro získání této role nutná buď znalost vstupního klíče ke kurzu (obr. 27) a nebo musí být tato role přidělena *vlastníkem* či *redaktorem* kurzu. Informace o přidělení této role je uložena v databázi, a proto je při příštím vstupu do kurzu uživateli vstup umožněn bez ověřování klíče.

Role ve vybraném kurzu:
Bez přístupu

Zadejte vstupní klíč:

Vstup pouze po zadání klíče!

Obrázek 27: Ověření vstupního klíče

Pro veřejné kurzy je role *čtenář přihlášený* přidělována při vstupu a informace o přidělení role není ukládána do databáze.

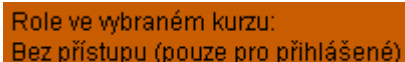
Role ve vybraném kurzu:
Čtenář

Obrázek 28: Čtenář přihlášený

Čtenář nepřihlášený

Při vstupu nepřihlášeného uživatele do kurzu je velmi důležité, jakou úroveň zabezpečení má kurz nastavenu (viz kapitola 2.1.1.2 na str. 28 – *úroveň zabezpečení kurzu*).

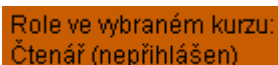
Nepřihlášenému uživateli nebude nikdy umožněno vstoupit do soukromého kurzu. V soukromých kurzech tedy vždy získá roli *bez přístupu*.



Role ve vybraném kurzu:
Bez přístupu (pouze pro přihlášené)

Obrázek 29: Bez přístupu pro soukromý kurz

U veřejných kurzů je automaticky přidělena role *čtenář nepřihlášený* (až na situaci, která může nastat, pokud je některý z nadřazených kurzů soukromý). Z hlediska portálu se nepřihlášený a přihlášený čtenář neliší, ale v jednotlivých modulech mohou být aktivity, které jsou vázány na uživatelské jméno čtenáře, a ty potom budou nepřihlášenému čtenáři nepřístupné (více v části věnované jednotlivým modulům).

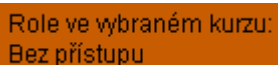


Role ve vybraném kurzu:
Čtenář (nepřihlášen)

Obrázek 30: Čtenář nepřihlášený

Bez přístupu

Uživateli s rolí *bez přístupu* nebude umožněn přístup k obsahu modulů. Toto odepření přístupu je řešeno na úrovni portálu, proto nemusí být začleněno do zabezpečení modulů samotných.



Role ve vybraném kurzu:
Bez přístupu

Obrázek 31: Bez přístupu

2.1.3.2 Práva uživatelů při práci s kurzy

Již v kapitole 2.1.1 Kurzy a zejména v části 2.1.1.2 Administrace kurzů jsme uvedli, jaké aktivity je možné provádět v souvislosti s kurzy. Zde se zaměříme na to, jaké uživatelské role jsou nutné k provádění jednotlivých akcí.

	<i>Vlastník</i>	<i>Redaktor</i>	<i>Příspěvatel</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Zobrazení seznamu kurzů	X	X	X	X	X	X
Zobrazení stromu kurzů	X	X	X	X	X	X
Práce s oblíbenými kurzy	X	X	X	X		
Zobrazení kurzů, ke kterým má uživatel právo	X	X	X	X		
Vstup do veřejného kurzu	X	X	X	X	X	
Vstup do soukromého kurzu	X	X	X	X		
Vytvoření nového (pod)kurzu	X	X				
Úprava vlastností kurzu	X	X				
Výmaz kurzu a podkurzů	X					
Výmaz kořenového kurzu						

Tabulka 1: Oprávnění pro práci s kurzy

V tabulce 1 jsou vyznačeny akce povolené uživatelům s danou rolí v kurzu. K těmto oprávněním se vážou další pravidla:

- *Redaktor*, který v kurzu založí „podkurz“, se stává vlastníkem vytvořeného „podkurzu“. Díky tomu je dobře decentralizována správa portálu.
- Uživateli s rolí *vlastník* kurzu je tato role přidělena i pro všechny podřízené kurzy (pro celý podstrom „podkurzů“). V důsledku toho jsou vlastníci kořenového kurzu portálu vlastníky všech kurzů portálu.
- Při vstupu do soukromého kurzu je přihlášenému uživateli přidělena role *čtenář přihlášený*, pokud správně zadá (nebo již v minulosti zadal) vstupní klíč, nebo pokud mu tato role byla v kurzu přidělena vlastníkem či redaktorem kurzu.

2.1.3.3 Správa uživatelů portálu

Další důležitou oblastí, ve které jsou nutné správcovské zásahy, jsou uživatelské účty v portálu. Změny hesel, blokování a rušení uživatelů, kteří porušují pravidla portálu a nebo naopak přidělení vyšší role uživateli, který se má podílet na vytváření obsahu kurzu – to jsou jen některé úkoly, které podrobněji rozebere tato kapitola.

Hlavní uživatel

Prvním uživatelem portálu, který v něm existuje od samotného začátku, je *hlavní uživatel*. Jeho ID v tabulce uživatelů je 1 (uživatelské jméno může být různé). Tento uživatel je na začátku vlastníkem kořenového kurzu a tudíž celého portálu.

Hlavní uživatel má v mnoha ohledech výjimečné postavení a liší se od ostatních vlastníků kořenového kurzu:

- Údaje o hlavním uživateli (jako například heslo) může měnit jen hlavní uživatel sám.
- Nikdo (ani sám hlavní uživatel) nemůže zbavit hlavního uživatele vlastnictví hlavního (kořenového) kurzu.
- Nikdo nemůže hlavního uživatele odstranit.
- Nikdo nemůže hlavního uživatele zablokovat tak, že by se nemohl přihlásit.

Správa všech uživatelů portálu

V této části se zaměříme na úkony spojené se správou uživatelských účtů. Správu všech uživatelů portálu může provádět pouze ten uživatel, který má **v kořenovém kurzu přidělenou roli *vlastník***. Tento uživatel je totiž, jak bylo ukázáno dříve, vlastníkem všech kurzů – neboli celého portálu. Správa všech uživatelů se aktivuje pomocí odkazu *Editace všech uživatelů portálu*, který se zobrazuje pouze v kořenovém kurzu.



Role ve vybraném kurzu:
Vlastník
[Skrýt editaci všech uživatelů portálu](#)

Upravit uživatele

přezdívka:

[Zobrazit všechny uživatele](#)
[Vytvořit nového uživatele](#)

Obrázek 32: Základní nabídka editace všech uživatelů

Správce má dvě možnosti, jak uživatele editovat. Může do formuláře (viz obrázek 32) zadat přezdívku (uživatelské jméno) uživatele, pokud ji zná – například na základě záznamu o absolvování testu, příspěvku v diskusi, ... Nebo může zobrazit všechny uživatele a zvolit, kterého chce editovat (viz Obrázek 33).

Zobrazuji všechny uživatele portálu				
Skrýt				
id	přezdivka	jméno	příjmení	blokován
5	testctenar	Tester	Čtenář	ne
4	testprispevateľ	Tester	Příspěvatel	ne
3	testredaktor	Tester	Redaktor	ano
2	testvlastník	Tester	Vlastník	ne
1	vyroubal	Michal	Vyroubal	ne

Obrázek 33: Seznam všech uživatelů portálu

Je možné editovat veškeré údaje, které zadal uživatel při registraci, kromě uživatelského jména. Navíc může správce uživatele zablokovat, čímž mu znemožní přihlášení k portálu. Uživatel však dále bude existovat v databázi, a proto budou jeho aktivity z minulosti nadále navázány na jeho přezdívkou. Některé uživatele (například redaktory, kteří jsou autory velké části výukového obsahu v modulech) je tedy lepší nemazat, ale pouze blokovat. Je-li totiž uživatel smazán, začnou být všechny jeho příspěvky buď zobrazovány bez autora, nebo s autorem *anonym* (záleží na jednotlivých modulech).

Pokud správce do polí pro změnu hesla nic nevyplní, zůstane uživateli jeho původní heslo nezměněno.

Editace uživatele:

Přezdivka: testredaktor

Jméno:

Příjmení:

e-mail:

Heslo:

Ověření hesla:

Zablokovat uživatele: ☒

Smazat uživatele

Obrázek 34: Formulář editace vlastností uživatele

Správce portálu může vytvořit uživatele portálu ručně – například pokud by chtěl studentům přiřadit uživatelská jména systematicky. Připomeňme však, že uživatelé se mohou do portálu registrovat sami bez účasti správce.

Vytvoření nového uživatele:

Vyplňte prosím následující formulář (přezdivka je povinná)

Přezdivka:

Jméno:

Příjmení:

e-mail:

Heslo:

Ověření hesla:

[Skrýt](#)

Obrázek 35: Vytvoření nového uživatele správcem portálu

Správa rolí uživatelů vybraného kurzu

Přejdeme ke správě rolí, které mají uživatelé v jednotlivých kurzech. Nejprve si ukážeme, které aktivity jsou přístupné uživatelům v jednotlivých rolích. Správa uživatelů kurzu je přístupná pouze uživatelům, kteří mají ve zvoleném kurzu roli *vlastník* nebo *redaktor*. Proto nebudou v následující tabulce uvedeny žádné další role.

		<i>Vlastník</i>	<i>Redaktor</i>
Přřadit kterémukoli uživateli roli	Vlastníka	X	
	Redaktora	X	
	Příspěvatele	X	
	Čtenáře	X	X
Odebrat v kořenovém kurzu roli vlastníka hlavnímu uživateli portálu			
Odebrat uživateli roli	Vlastníka	X	
	Redaktora	X	
	Příspěvatele	X	
	Čtenáře	X	X

Tabulka 2: Oprávnění pro správu uživatelů kurzu

Správa rolí uživatelů kurzu se aktivuje odkazem *Editace uživatelů tohoto kurzu*. Zobrazí se tabulka rolí a uživatelé, kteří mají v kurzu přidělené role. Jak vyplývá z tabulky 2 uvedené výše, forma zobrazení záleží na tom, zda přihlášený uživatel má roli *vlastník* (viz Obrázek 36) nebo *redaktor* (viz Obrázek 37).

Skrýt editaci uživatelů tohoto kurzu			
Pro vlastníka		Pro redaktora	
Vlastníci	Redaktori	Příspěvatele	Čtenáři
Přidat vlastníka	Přidat redaktora	Přidat příspěvatele	Přidat čtenáře
přezdívka: <input type="text"/> Přidat	přezdívka: <input type="text"/> Přidat	přezdívka: <input type="text"/> Přidat	přezdívka: <input type="text"/> Přidat
id nickname příjmení jméno 2 testvlastnik Vlastník Tester 1 vyroubal Vyroubal Michal	id nickname příjmení jméno 3 testredaktor Redaktor Tester	id nickname příjmení jméno 4 testprispevatele Příspěvateľ Tester	id nickname příjmení jméno 5 testctenar Čtenář Tester

Obrázek 36: Základní nabídka editace rolí uživatelů kurzu – pohled vlastníka

[Skrýt editaci uživatelů tohoto kurzu](#)

<i>Pro redaktora</i>			
Čtenáři			
Přidat čtenáře			
přezdívk:	<input type="text"/>	<input type="button" value="Přidat"/>	
<i>id</i>	<i>nickname</i>	<i>příjmení</i>	<i>jméno</i>
5	testctenar	Čtenář	Tester

Obrázek 37: Základní nabídka editace rolí uživatelů kurzu – pohled redaktora

Uživatel s rolí *redaktor* může plně spravovat pouze čtenáře kurzu. Tuto možnost můžeme využít v situaci, kdy nechceme registrovat uživatele do soukromého kurzu pomocí vstupního klíče, ale ručně.

Vybranému uživateli, který již nějakou roli v aktuálním kurzu přiřazenu má, je možné tuto roli odebrat nebo ji libovolně povýšit či ponížit (viz Obrázek 38). Jedinou výjimkou je nemožnost odebrat hlavnímu uživateli roli *vlastník* v kořenovém kurzu.

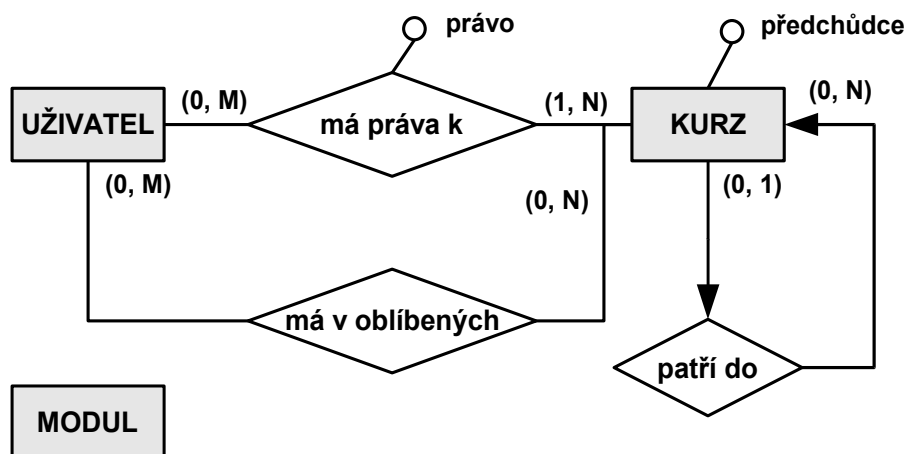
[Skrýt podrobnosti o uživateli](#)

přezdívk	jméno	příjmení
testvlastnik	Tester	Vlastník
odebrat roli vlastníka		
Změnit roli na: redaktora		
příspěvatele		
čtenáře		
Skrýt editaci uživatelů tohoto kurzu		
Vlastníci		
Přidat vlastníka		
přezdívk:	<input type="text"/>	<input type="button" value="Přidat"/>
<i>id</i>	<i>nickname</i>	<i>příjmení</i> <i>jméno</i>
2	testvlastnik	Vlastník Tester
1	vyroubal	Vyroubal Michal

Obrázek 38: Editace role pro zvoleného uživatele

2.2 Databáze

Na základě konvencí rozebraných v kapitole 1.2.3 nyní zakreslíme ER diagram části databáze, která funkčně přináleží portálu.



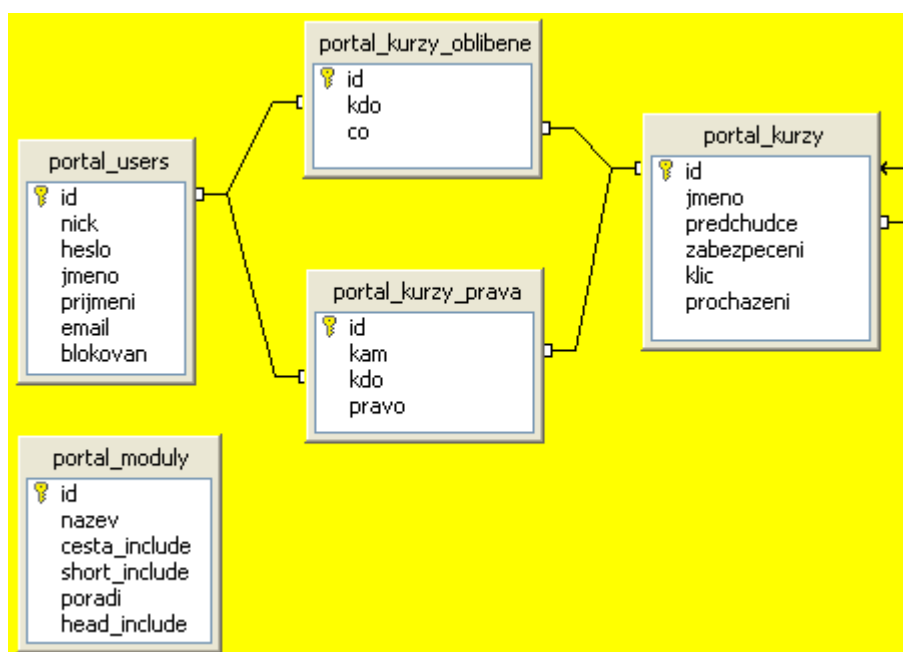
Obrázek 39: ER diagram databáze portálu

Entitní typ *KURZ* je propojen rekurentní relací, díky které je možné uspořádat kurzy do stromu. Každý kurz (kromě kořenového kurzu) obsahuje v atributu *předchůdce*, který je cizím klíčem, odkaz na nadřazený kurz. Tyto (pod)kurzy tedy mají členství v relaci povinné (neboli vyjádřeno zápisem MIN-MAX IO je podřízený kurz (1, 1) vzhledem ke kurzu nadřazenému). Entita kořenového kurzu není součástí žádné relace jakožto podřízený kurz, ale může být (pokud je v portálu více než jeden kurz) součástí relací, kde vystupuje v roli kurzu nadřazeného, neboli *předchůdce*. Obecně je tedy tato relace pro podřízené kurzy typu (0, 1) a pro nadřazené kurzy typu (0, N) – ke kurzu nemusí existovat žádný podřízený kurz.

Relační vztah *má práva k* mezi entitními typy *KURZ* a *UŽIVATEL* je v tomto diagramu popsán tak, že členství entity *KURZ* je povinné. Toto povinné členství budeme však vyžadovat pouze pro kořenový kurz portálu, kde musí být alespoň jeden uživatel definován jako *vlastník*. Pro podřízené kurzy již nebude členství v relaci povinné, protože vlastník určitého kurzu je automaticky vlastníkem i všech podřízených kurzů ve stromové struktuře kurzů a tudíž přiřazením vlastníka ke kořenovému kurzu portálu současně určujeme vlastníka všech kurzů.

Atribut *právo* je definován u relace *má práva k*, díky čemuž může mít uživatel pro různé kurzy různá práva.

Entita *MODUL* není v relaci s ostatními entitami. Slouží pouze jako část rozhraní portálu pro přidávání modulů. Význam tohoto entitního typu bude vysvětlen v kapitole 2.4.2 *Tabulka pro registraci modulu v portálu*.



Obrázek 40: Znárodnění tabulek, atributů a relací mezi tabulkami

Tabulky *portal_kurzy_prava* a *portal_kurzy_oblibene* (viz Obrázek 40) představují průnikové entitní typy vzniklé dekompozicí relací *má práva k* a *má v oblíbených* (viz Obrázek 39) mezi entitami UŽIVATEL (tabulka *portal_users*) a KURZ (tabulka *portal_kurzy*). Entitu MODUL zde zastupuje tabulka *portal_moduly*.

Data

Databáze portálu od začátku obsahuje záznamy:

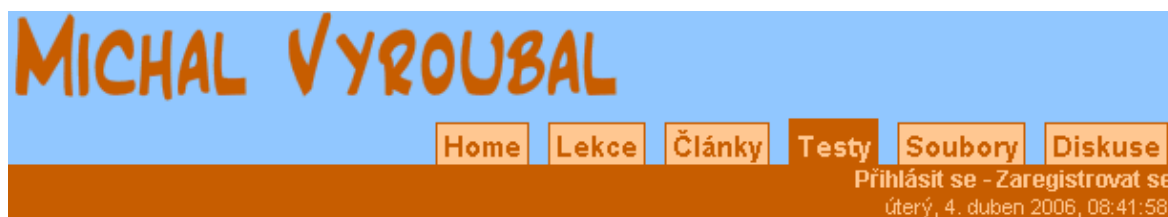
- tabulka *portal_users* – je vytvořen hlavní uživatel portálu
- tabulka *portal_kurzy* – je vytvořen hlavní (kořenový) kurz
- tabulka *portal_kurzy_prava* – je vytvořen záznam, který přiřazuje hlavnímu uživateli vlastnická práva pro hlavní kurz
- tabulka *portal_moduly* – jsou řádně zaregistrovány všechny moduly (bude dále rozebráno)

2.3 PHP kód

Realizaci portálu a konkrétní řešení pomocí PHP nebudeme rozebírat samostatně, ale tam, kde to bude účelné pro lepší pochopení nebo hlubší porozumění probírané problematiky, budeme PHP kód začleňovat přímo do textu. Navíc je v příloze na CD k dispozici kompletní zdrojový kód portálu.

2.4 Rozhraní pro moduly

Portál poskytuje rozhraní, přes které je možné k němu připojit moduly. Součástí této diplomové práce je i vytvoření šesti základních modulů. Nyní popíšeme rozhraní portálu pro moduly tak, aby bylo možné na základě uvedeného popisu vytvořit nový modul pro portál a připojit jej k portálu.



Obrázek 41: Nabídka modulů

2.4.1 Informace o umístění portálu a aktuální URL

Základní adresa URL, na které je portál umístěn, je uložena ve dvou konstantách (soubor `include/config.php`):

<code>HOME_URL</code>	URL nešifrovaného portálu <ul style="list-style-type: none"> například: <code>http://server/home_dir/</code>
<code>HOME_SSL</code>	URL portálu zabezpečeného pomocí šifrování <ul style="list-style-type: none"> například: <code>https://server/home_dir/</code>

K přechodu mezi šifrovaným a nešifrovaným spojením dochází při přihlašování a odhlásování uživatele. Při tomto přechodu se zkopíruje hodnota příslušné (jedné z výše uvedených) konstanty do session proměnné `$_SESSION["HOME_URL"]`. Pomocí této session proměnné je možné vytvářet v modulu odkazy bez rozlišování použitého přenosového protokolu.

Kód, který nastavuje a používá `$_SESSION["HOME_URL"]` při **přihlášení** (soubor `users/login_out.php`):

```
$_SESSION["HOME_URL"] = HOME_SSL;
header ("Location: " . $_SESSION["HOME_URL"] .
        "index.php?modul=0&otec=1");
```

Kód, který nastavuje a používá `$_SESSION["HOME_URL"]` při **odhlášení** (soubor `users/login_out.php`):

```
$_SESSION["HOME_URL"] = HOME_URL;
header ("Location: " . $_SESSION["HOME_URL"] .
        "index.php?modul=0&otec=1");
```

```
$_SESSION['HOME_URL']
```

Aktuální kořenové umístění portálu

- obsahuje hodnotu jedné z konstant
`HOME_URL` nebo `HOME_SSL`

Při práci s portálem se k základní URL připojují GET hodnoty. Například hned na začátku se připojují údaje o zvoleném modulu (proměnná `modul`) a kurzu (proměnná `otec`):

```
| ?modul=1&otec=1
```

Tento řetězec je postupně sestavován a uchováván v proměnné `$url_hodnoty`. Při další práci se tento řetězec prodlužuje o další a další proměnné. Z toho pro modul vyplývá jedna výhoda a jedna povinnost. Výhodou je možnost použít tuto proměnnou pro tvorbu odkazů a zůstat tak nezávislým na kurzu. Povinností je uvedenou proměnnou dále doplňovat o důležité hodnoty. Na konci běhu skriptu je tato proměnná zobrazena na stránce (viz Obrázek 42).

Adresa tohoto příspěvku: http://server6/diplomka/index.php?modul=3&otec=6&clanky_show=1&clanky_clanek=2

Obrázek 42: Zobrazení proměnné `$url_hodnoty` na konci skriptu

```
$url_hodnoty
```

GET hodnoty z URL zobrazované stránky


- vytváří se postupně v průběhu celého skriptu

Sestavení odkazu v kódu modulu potom může vypadat například takto (jde o příklad z modulu *Články* – zobrazení článku; pole `$radek` zde obsahuje informace o článku):

```
echo "<a href=\"\"
    .$_SESSION["HOME_URL"]
    ."index.php"
    .$url_hodnoty
    ."&clanky_show=1&clanky_clanek=".$_radek[0]
    .\">Celý článek</a>";
```

2.4.2 Tabulka pro registraci modulu v portálu

Pro správné zaregistrování modulu v portálu je nutné přidat záznam do tabulky *portal_moduly*. Proto si nyní popíšeme význam jednotlivých atributů entitního typu *MODUL*, respektive polí tabulky *portal_moduly*.

portal_moduly	
	id
	název
	cesta_include
	short_include
	poradi
	head_include

Obrázek 43: Tabulky pro registraci modulu v portálu

id	Primární klíč <ul style="list-style-type: none"> • automaticky generován • není nutné zadávat
nazev	Název modulu <ul style="list-style-type: none"> • tímto názvem bude modul označen v záložkách modulů (viz Obrázek 41) • nejlépe jednoslovný
cesta_include	Cesta k hlavnímu souboru modulu <ul style="list-style-type: none"> • tento soubor se bude includovat do portálu
short_include	Cesta k souboru se zkrácenou informací o modulu <ul style="list-style-type: none"> • tento soubor je includován základním modulem <i>Home</i> • jeho funkce jsou plně v kompetenci tvůrce modulu • může například informovat o počtu objektů v modulu příslušejících vybranému kurzu
poradi	Pořadí modulů <ul style="list-style-type: none"> • podle tohoto pořadí se budou zobrazovat záložky modulů v portálu ve směru zleva doprava (viz Obrázek 41 na str. 43)
head_include	Cesta k souboru s hlavičkou <ul style="list-style-type: none"> • tento soubor je includován při sestavování hlavičky HTML dokumentu (<head> ... </head>) souborem head.php

Příklad vložení záznamu pro modul *Články*:

```
INSERT INTO `portal_moduly`
  (`nazev`, `cesta_include`,
   `short_include`, `poradi`, `head_include`)
VALUES
  ('Články', 'moduly/clanky/hlavni.php',
   'moduly/clanky/short.php', 2, 'moduly/clanky/head.php');
```

Uvedeným atributům odpovídají session proměnné, do kterých se načítají hodnoty atributů pro jednotlivé moduly. O zmíněných session proměnných je pojednáno v následující kapitole.

2.4.3 Informace o modulech

Informace o všech modulech portálu se načítají z tabulky *portal_moduly* a poté jsou uloženy do session proměnných. Informace o modulech se načítají při prvním načtení portálu a potom pokaždé, když je vybrán modul *Home*.

Základní část kódu načítajícího informace o modulech (soubor `index.php`):

```
$dotaz = "SELECT * FROM `".TAB_MODULEY."` ORDER BY `poradi`";
$moduly = $db->sql_dotaz($dotaz);
$i = 0;
foreach($moduly as $modul) {
    session_register("modul".$i."id");
    $_SESSION["modul".$i."id"] = $modul[0];
    session_register("modul".$_SESSION["modul".$i."id"]."name");
    session_register("modul".$_SESSION["modul".$i."id"]."path");
    session_register("modul".$_SESSION["modul".$i."id"]."short");
    session_register("modul".$_SESSION["modul".$i."id"]."head");
    $_SESSION["modul".$_SESSION["modul".$i."id"]."name"] = $modul[1];
    $_SESSION["modul".$_SESSION["modul".$i."id"]."path"] = $modul[2];
    $_SESSION["modul".$_SESSION["modul".$i."id"]."short"] = $modul[3];
    $_SESSION["modul".$_SESSION["modul".$i."id"]."head"] = $modul[5];
    ++$i;
}
session_register("moduly");
$_SESSION['moduly'] = --$i;
```

Význam jednotlivých session proměnných:

`$_SESSION['moduly']`

Nejvyšší pořadí modulu

- hodnota `$_SESSION['moduly'] + 1` představuje počet modulů

`$_SESSION["modul".$i."id"]`

Identifikátor modulu

- jde o primární klíč v tabulce modulů
- proměnná `$i` odpovídá pořadí modulů, ve kterém se zobrazují

V následujících řádcích zastupuje `id_modulu`
zápis session proměnné `$_SESSION["modul".$i."id"]`

`$_SESSION["modul".id_modulu."name"]`

Název modulu

- zobrazuje se v různých situacích (například na záložkách modulů)

`$_SESSION["modul".id_modulu."path"]`

Cesta k hlavnímu souboru modulu

- tento soubor se bude includovat do portálu

`$_SESSION["modul".id_modulu."short"]`

Cesta k souboru se zkrácenou informací

- tento soubor je includován modulem *Home*
- může poskytovat například informaci o počtu uveřejněných objektů v modulu

```
$_SESSION
["modul".id_modulu."head"]
```

Cesta k souboru s hlavičkou

- tento soubor je includován při sestavování hlavičky HTML dokumentu (<head> ... </head>)
- může obsahovat například odkaz na definice css stylů používaných pouze daným modulem

Příklad kódu, který používá session proměnnou

`$_SESSION["modul".id_modulu."path"]` je na straně 49.

2.4.4 Zvolený modul

Aktuálně zvolený modul se načítá z asociativního pole `$_GET`, neboli z hodnot předávaných pomocí URL. Nejprve uvedeme klíčové místo kódu, který provádí nastavení zvoleného modulu (soubor `index.php`):

```
if(!session_is_registered("set_modul"))
    session_register("set_modul");
if(isset($_GET['modul']) && ((int)$_GET['modul'] >= 1))
    $_SESSION["set_modul"] = (int)$_GET['modul'];
else //pokud není zvoleno nic, nastaví 1. modul - Home
    $_SESSION["set_modul"] = "1";
```

Z tohoto kódu vyvstává další důležitá session proměnná:

`$_SESSION['set_modul']` **Identifikátor zvoleného modulu**

- obsahuje primární klíč zvoleného modulu z tabulky modulů (*portal_moduly*)

Příklad kódu, který používá session proměnnou `$_SESSION['set_modul']` je na straně 49.

2.4.5 Zvolený kurz

Aktuálně zvolený kurz je rovněž uchovávan v session proměnné. Modul potřebuje tuto hodnotu zejména k ověřování oprávněnosti přístupu, protože veškerá práva uživatele jsou vázána na kurz.

`$_SESSION['selected_tema']` **Identifikátor zvoleného kurzu**

- jde o primární klíč v tabulce kurzů (*portal_kurzy*)

2.4.6 Uživatelé portálu

Informace o přihlášeném uživateli jsou ukládány v session proměnných.

`$_SESSION['usr_id']`

Identifikátor uživatele

- primární klíč v tabulce uživatelů

`$_SESSION['usr_name']` **Přezdívk**a (uživatelské jméno) uživatele

- odpovídá obsahu pole *nick* v tabulce *portal_users*

V okamžiku přihlášení uživatele se provedou příkazy (pole `$user` obsahuje informace načtené z tabulky *portal_users* – soubor `users/login_out.php`):

```
session_register("usr_id");
session_register("usr_name");
$_SESSION['usr_id'] = $user[0][0];
$_SESSION['usr_name'] = $user[0][1];
```

Při odhlášení se provedou příkazy (soubor `users/login_out.php`):

```
session_unregister("usr_id");
session_unregister("usr_name");
```

Proto je možné pro zjištění, zda je aktuální uživatel přihlášen nebo je anonymní, použít podmínky:

<code>if(isset(\$_SESSION['usr_id']))</code>	Pravda, pokud uživatel je přihlášen.
<code>if(!isset(\$_SESSION['usr_id']))</code>	Pravda, pokud uživatel není přihlášen.

2.4.7 Přístupová práva a role

Velmi důležitou informací, kterou vyhodnocuje portál a předává ji modulu, je oprávnění a role, kterou uživatel získal ve zvoleném kurzu. Pro předání této informace se používá několik session proměnných:

<code>\$_SESSION['ma_prava_tema']</code>	Přístup k vybranému kurzu <ul style="list-style-type: none"> • tato proměnná je rovna 1 právě tehdy, když uživatel má právo zobrazit obsah aktuálně zvoleného kurzu • jinak je rovna 0
<code>\$_SESSION['ma_prava_tema_final']</code>	Přístup s ohledem na nadřazené kurzy <ul style="list-style-type: none"> • tato proměnná je rovna 1 právě tehdy, když uživatel získal oprávnění procházet všemi nadřazenými kurzy až ke kořenovému kurzu • jinak je rovna 0 • příklad: Pokud je některý z nadřazených kurzů soukromý a má zakázané procházení, bude pro podřízené kurzy přidělena této proměnné hodnota 0 (více viz strana 28 – <i>povolit procházení kurzem</i>).

Výše uvedené session proměnné jsou použity v souboru `index.php` pro vyhodnocení, zda má být includován hlavní soubor modulu (význam session proměnných týkajících se modulů je v kapitole 2.4.3 *Informace o modulech* na straně 45 a v kapitole 2.4.4 *Zvolený modul* na straně 47):

```
if($_SESSION["ma_prava_tema_final"] == "1"
    && $_SESSION["ma_prava_tema"] == "1") {
    if(isset($_SESSION["set_modul"])){
        if(isset($_SESSION["modul".$_SESSION["set_modul"].".path"]))
            include $_SESSION["modul".$_SESSION["set_modul"].".path"];
        else
            echo "Takový modul neexistuje";
    }
} else
    echo "Přístup zamítnut";
```

Pokud tedy jedna z uvedených session proměnných je různá od hodnoty 1, není proveden include modulu. Je-li include modulu proveden, bude muset modul znát roli, která byla uživateli v daném kurzu přidělena. Seznam rolí je definován pomocí konstant v souboru `include/config.php`:

```
define("VLASTNIK", '10');
define("REDAKTOR", '20');
define("PRISPEVATEL", '30');
define("CTENAR", '40');
```

Hodnoty těchto konstant jsou ukládány do databáze do tabulky *portal_kurzy_prava* do pole *pravo* (viz obrázky 39 a 40), a proto by nebylo dobré konstanty za chodu portálu předdefinovávat. Konstanty představují pouze symbolické pojmenování hodnot vedoucí k vyšší přehlednosti kódu.

Po vyhodnocení přístupových práv je role uživatele uložena do session proměnné:

`$_SESSION['tema_pravo']` **Role uživatele ve zvoleném kurzu**

Příklad kódu, který v modulu zjišťuje roli uživatele (test, zda je aktuálně přihlášený uživatel vlastníkem nebo redaktorem kurzu):

```
if($_SESSION["tema_pravo"] == VLASTNIK
    || $_SESSION["tema_pravo"] == REDAKTOR)
```

Modul je tedy úspěšně zaregistrován v portálu a tvůrce modulu může účinně využívat služby, které portál poskytuje. Základní úloha portálu je naplněna a my se budeme dále věnovat modulům obecně a následně konkrétním modulům, které byly pro portál vytvořeny v rámci této diplomové práce.

3 Moduly







Stavebními kameny e-learningového portálu jsou moduly. Moduly se začleňují do portálu a tvoří společně jednotný celek. Rozhraní, které poskytuje portál modulům, jsme již rozebrali v předchozích kapitolách. Nyní se budeme věnovat vlastnostem a rysům společným pro více modulů a dále jednotlivým konkrétním modulům.

3.1 Společné vlastnosti modulů

Dříve než se začneme věnovat jednotlivým modulům a jejich specifickým funkcím a vlastnostem, ukážeme si, co mají moduly společného. Funkce modulů, kterými se nyní budeme zabývat, nemusí být vždy ve všech modulech, a proto budeme uvádět výčet modulů, které rozebíranou funkčnost poskytují.

3.1.1 Ikony pro administraci

Ve všech modulech se vyskytují ikony sloužící pro administraci modulu. Tyto ikony jsou pro zvýšení přehlednosti práce jednotné. Ikony jsou uloženy v adresáři `pictures`, kde je možné je zaměňovat.

<i>Ikona</i>	<i>Jméno souboru</i>	<i>Význam</i>
	<code>ikona_delete.png</code>	Smazání objektu
	<code>ikona_edit.png</code>	Editace vlastností objektu
	<code>ikona_up.png</code>	Přesunutí objektu výše
	<code>ikona_down.png</code>	Přesunutí objektu níže
	<code>ikona_visible.png</code>	Objekt je uveřejněn
	<code>ikona_unvisible.png</code>	Objekt není uveřejněn

Další možností, jak změnit tyto ikony je předefinování konstant definovaných v souboru `include/config.php`. Definice používání ikony vypadá například takto:

```
define("TEXT_DEL",
    '');
```

3.1.2 Změna pořadí objektů

Změnu pořadí objektů je možné provádět v modulech *Články*, *Home*, *Soubory*, *Diskuse*, *Testy* i *Lekce*. Mluvme však rovnou o modulu *Články*, který nám poslouží jako univerzální příklad v této i dalších kapitolách.

Články se řadí v tom pořadí, jak byly vytvářeny, takže nahoře v seznamu článků je nejstarší článek a na konci je článek nejmladší. Uživatel, který si prohlíží seznam článků,

může toto pořadí obrátit pomocí odkazů *Seřadit články: vzestupně – sestupně*. Někdy je však nutné pořadí článků promíchat, aby lépe odpovídalo didaktickým a metodickým vazbám uvnitř probíraného učiva. Toto přehazování se provádí pomocí ikon:



Přesunutí objektu výše



Přesunutí objektu níže

Práva

Pro změnu pořadí článků musí mít uživatel v kurzu roli *vlastník* nebo *redaktor*. *Příspěvatel* a samozřejmě ani *čtenář* nemohou měnit pořadí.

Databáze

Jediným požadavkem na atributy v databázi je, aby v hlavní tabulce modulu existovalo pole *poradi* (viz Obrázek 44 na straně 53).

Kód při vytvoření nového objektu

Při vytvoření nového článku se nejprve zjistí, jaké je aktuálně nejvyšší pořadí článku ve zvoleném kurzu a přiřadí se pořadí o jedna vyšší:

```
$dotaz = "SELECT max(`poradi`) FROM `".TAB_CLANKY
        ."` WHERE `tema`='". $id_tema. "'";
$max_poradi = $this->sql_dotaz($dotaz);
if($max_poradi[0][0] > 0) $max_poradi = $max_poradi[0][0];
else $max_poradi = 0;
```

Následující SQL příkaz INSERT je redukován pouze na ukázkou vložení pořadí článku:

```
$dotaz = "INSERT INTO `".TAB_CLANKY."` (`poradi`) VALUES ('"
        . ++$max_poradi . "')";
$this->sql_dotaz($dotaz);
```

Kód při změně pořadí objektů

Při změně pořadí článků jde v podstatě o prohození hodnot atributu *poradi*. Posouvám-li tedy článek nahoru, prohodím hodnotu *poradi* posouvaného článku, a článku, který má nejbližší nižší pořadí. Primární atributy pro tyto dvě entity jsou zjišťovány ve chvíli, kdy se články zobrazují. (Následující kód je zjednodušen tak, že ukazuje pouze zobrazení ikon pro změnu pořadí článků.)

```
for($i = 0; $i < sizeof($clanky); $i++){
    $radek = $clanky[$i];
    if($i > 0)
        echo "<a href=\"".$_SESSION["HOME_URL"]."index.php". $url_hodnoty
            . "&clanky_swap=1&clanky_clanek=" . $radek[0]
            . "&clanek_swap_id=" . $clanky[$i-1][0]. "\">".TEXT_UP."</a>";
    if($i < sizeof($clanky) - 1)
        echo "<a href=\"".$_SESSION["HOME_URL"]."index.php". $url_hodnoty
            . "&clanky_swap=1&clanky_clanek=" . $radek[0]
            . "&clanek_swap_id=" . $clanky[$i+1][0]. "\">".TEXT_DOWN."</a>";
}
```

Pomocí URL se předávají dvě hodnoty – `clanky_clanek` je hodnota primárního atributu daného článku a `clanek_swap_id` je hodnota primárního atributu článku, který má nejbližší nižší (respektive vyšší) pořadí. S těmito parametry je volána samotná metoda pro přehození článků:

```
$new_clanek->swap_clanky($_SESSION['selected_tema'],
                        $_GET['clanky_clanek'],
                        $_GET['clanek_swap_id']);
```

Samotné přehození potom probíhá takto:

```
function swap_clanky($id_tema, $id_clanek_1, $id_clanek_2){
    //Overeni, zda zvolene clanky patri do aktualniho tematu
    if($this->overeni_platnosti_pristupu(TAB_CLANKY, "tema",
                                        "id", $id_clanek_1)

        != $id_tema
    ||
        $this->overeni_platnosti_pristupu(TAB_CLANKY, "tema",
                                        "id", $id_clanek_2)

        != $id_tema)
        die("Jeden ze článků nepatří do tohoto kurzu");
    //Nacteni stavajiciho poradi
    $dotaz = "SELECT `poradi` FROM `".TAB_CLANKY
            ."` WHERE `id`='". $id_clanek_1.'" LIMIT 1";
    $poradi_1 = $this->sql_dotaz($dotaz);
    $dotaz = "SELECT `poradi` FROM `".TAB_CLANKY
            ."` WHERE `id`='". $id_clanek_2.'" LIMIT 1";
    $poradi_2 = $this->sql_dotaz($dotaz);
    if(sizeof($poradi_1) > 0 && sizeof($poradi_2) > 0){
        $dotaz = "UPDATE `".TAB_CLANKY."` SET "
            . "`poradi`='". $poradi_2[0][0] . "' "
            . "WHERE `id`='". $id_clanek_1 . "' LIMIT 1";
        $this->sql_dotaz($dotaz);
        $dotaz = "UPDATE `".TAB_CLANKY."` SET "
            . "`poradi`='". $poradi_1[0][0] . "' "
            . "WHERE `id`='". $id_clanek_2 . "' LIMIT 1";
        $this->sql_dotaz($dotaz);
    }else
        echo "Články, které chcete přehodit, neexistují.";
}
```

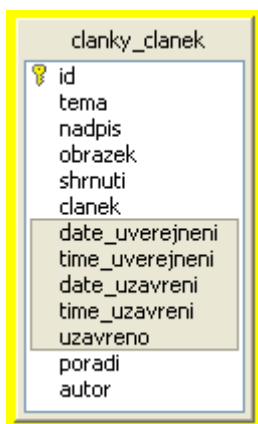
3.1.3 Načasování uveřejnění a stažení

Začněme příkladem: Učitel si vytvoří v portálu den předem test, ale nechce, aby tento test byl studentům přístupný. Test by se měl pro studenty v portálu „objevit“ až v tu chvíli, kdy ho budou mít za úkol vypracovat. Po stanoveném čase by se měl test opět pro studenty v portálu „schovat“, aby nebyl přístupný.

Pro splnění výše uvedeného požadavku je v modulech *Články*, *Home*, *Soubory*, *Diskuse*, *Testy* i *Lekce* zahrnuta funkce načasování uveřejnění a stažení objektu. Její principy a fungování si ukážeme opět na příkladu modulu *Články*.

Databáze

Hlavní entita modulu má atributy *date_uverejneni*, *time_uverejneni*, *date_uzavreni*, *time_uzavreni* a *uzavreno* (viz Obrázek 44), které slouží pro vyhodnocení, zda je možné objekt zobrazit.



Obrázek 44: Atributy pro načasování uveřejnění a stažení

Uživatelské role

Načasování se týká pouze uživatelů s rolí *čtenář* – budou objekt vidět právě tehdy, když je uveřejněn. Uživatelé s rolí *vlastník*, *redaktor* a *příspěvatel* mohou vidět i neuveřejněné objekty daného kurzu. (*Příspěvatel* vidí i ty neuveřejněné objekty, jejichž autorem je jiný uživatel, aby měl přehled, jaké informace jsou již v kurzu připravené. Nebude tak docházet k duplicitám a zbytečné práci.)

Pokud má uživatel navíc práva editovat objekt (má roli *vlastník* či *redaktor* a nebo je *příspěvatelem* a jde o objekt, který sám vytvořil), bude se signalizovat pomocí ikony, zda byl objekt uveřejněn či nikoli:



Objekt je uveřejněn



Objekt není uveřejněn

Nastavení uveřejnění a stažení

Při vytváření objektu (například článku) je ve vytvářecím formuláři část věnovaná načasování. Výchozí *datum a čas uveřejnění* je přednastaveno na aktuální hodnoty, takže pokud se v tomto formuláři nic nemění, bude článek okamžitě po odeslání formuláře uveřejněn a tedy přístupný.

Změnu *data uveřejnění* a zadání *data uzavření* (stažení) je možné provést, je však nutné dodržovat zápis

- data ve tvaru *rok–měsíc–den* a
- času ve tvaru *hodina:minuta:sekunda*.

Příznak *nepřístupnost článku* má vyšší prioritu při vyhodnocování než *datum uveřejnění* a *uzavření*. Je-li tedy „zaškrtnut“, článek nebude uveřejněn a to bez ohledu na nastavení data a času uveřejnění a stažení.

Datum uveřejnění:	2006-01-13
Čas uveřejnění:	08:10:22
Datum uzavření:	
Čas uzavření:	
Nepřístupnost článku:	<input type="checkbox"/> Tento článek je nepřístupný

Obrázek 45: Nastavení uveřejnění a stažení

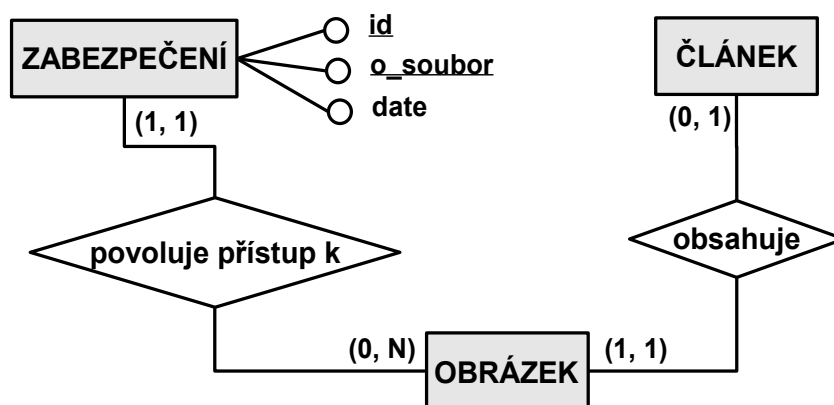
Článek je uveřejněn právě tehdy, když aktuální datum a čas je větší než *datum a čas uveřejnění* a současně je aktuální datum a čas menší než *datum a čas uzavření* a současně není aktivován příznak *nepřístupnost článku*.

3.1.4 Připojené obrázky

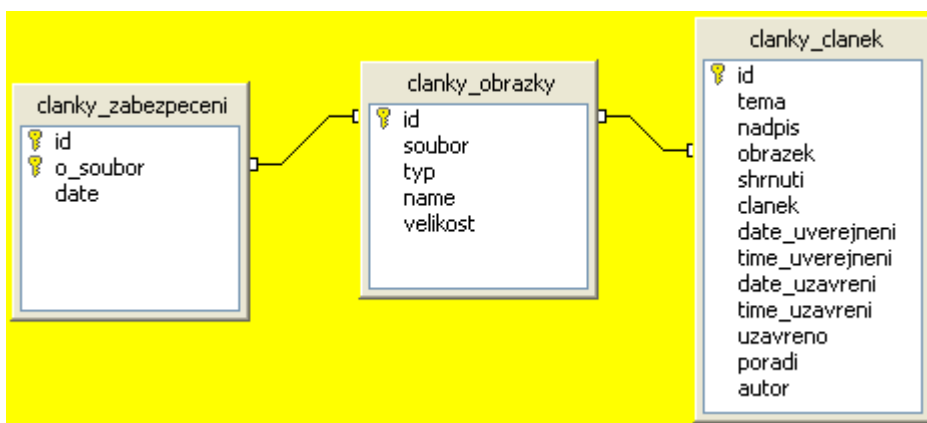
V některých modulech (jde o moduly *Články*, *Home* a *Testy*) je možné k objektům připojovat obrázky, které jsou ukládány do databáze. Nyní si ukážeme, jak probíhá práce s těmito připojenými obrázky. Jako příklad nám opět poslouží modul *Články*.

Databáze

Část databáze, která slouží pro uchovávání a zabezpečení připojených obrázků má v jednotlivých modulech v podstatě stále stejnou strukturu. Diagramy popisující tuto databázi jsou na obrázcích 46 a 47.



Obrázek 46: ER diagram části databáze pro připojení obrázku



Obrázek 47: Znáznornění tabulek, atributů a relací mezi tabulkami

Pro entitu *ZABEZPEČENÍ* je definován skupinový klíčový atribut tvořený atributy *id* a *o_soubor*. Atribut *id* obsahuje jako hodnotu hash (vytvořený funkcí md5) identifikátoru session. Atribut *o_soubor* je cizím klíčem (viz Obrázek 47). O účelu tohoto entitního typu je více pojednáno v části *Zabezpečení připojeného obrázku* na straně 56.

Členství entity objektu (v našem případě entity *ČLÁNEK*) je v relaci *obsahuje* s entitou *OBRAZEK* nepovinné, což umožňuje mít objekty, ke kterým není připojen obrázek. Kardinalita této relace je tedy 1:1, neboli k danému objektu může být připojen nejvýše jeden obrázek.

Připojení obrázku

K objektu je možné připojit obrázek pomocí formulářového pole, které je zahrnuto do formuláře pro vytvoření objektu.

Obrázek 48: Část formuláře pro připojení obrázku

Změna a odstranění připojeného obrázku

Pokud již k objektu byl připojen obrázek, je možné jej změnit tím, že se připojí obrázek jiný. Pokud je zaškrtnuto *Odebrat obrázek ze článku* (viz Obrázek 49) dojde ke zrušení vazby objektu na obrázek a k vymazání obrázku z databáze.



Obrázek 49: Část formuláře pro práci s vloženým obrázkem

Zabezpečení připojeného obrázku

Vytvoření záznamu v tabulce zabezpečení provádí metoda `zaznamenej_soubor` třídy `obrazky` (jméno této třídy se v jednotlivých modulech liší – v našem modelovém modulu *Články* se jmenuje `clanky_obrazky` a je v souboru `moduly/clanky/class_obrazky.php`):

```
function zaznamenej_soubor($tabulka, $soubor_id){
    $sid = md5(session_id()); //hash session id
    $dotaz = "SELECT `id` FROM `".$tabulka
        . "` WHERE `id`='".$sid."' "
        . "AND `o_soubor`='".$soubor_id."' LIMIT 1";
    $vysledek = $this->sql_dotaz($dotaz);
    if(sizeof($vysledek) > 0){
        //Pokud jiz dane zabezpeceni pro session existuje > UPDATE data
        $dotaz = "UPDATE `".$tabulka."` SET "
            . "`date`='".date("Y-m-d H:i:s") . "' "
            . "WHERE `id`='".$sid."' "
            . "AND `o_soubor`='".$soubor_id."' LIMIT 1";
    }else{
        //Pokud dane zabezpeceni pro session neexistuje, vytvori se
        $dotaz = "INSERT INTO `".$tabulka."` "
            . "(`id`, `o_soubor`, `date`) VALUES ("
            . $sid . ", "
            . $soubor_id . ", "
            . date("Y-m-d H:i:s") . ")";
    }
    $this->sql_dotaz($dotaz);
    //Odstraneni starych zabezpeceni z tabulky zabezpeceni
    $dotaz = "DELETE FROM `".$tabulka
        . "` WHERE `date`<="
        . date("Y-m-d H:i:s",
            mktime(
                date("H"), date("i")-15, date("s"),
                date("m"), date("d"), date("Y")))
        . "'";
    $this->sql_dotaz($dotaz);
}
```


Tato metoda se aplikuje ve chvíli, kdy je obrázek na stránce zobrazován. Vstupní parametry jsou `$tabulka` – jméno tabulky, ve které jsou uloženy záznamy zabezpečení (*clanky_zabezpeceni*) – a `$soubor_id` – hodnota primárního klíče z tabulky obrázků.

Nejprve se zjistí, zda není v tabulce zabezpečení záznam, který by opravňoval danou session zobrazit obrázek. Pokud se takový záznam nalezne, je jeho atribut *date* aktualizován na aktuální datum a čas. Jestliže žádný takový záznam nalezen není, tak se vytvoří. Nakonec metoda projde tabulku zabezpečení a vymaže všechna oprávnění starší než 15 minut, čímž se zamezí trvalému přidělení oprávnění a nekontrolovatelnému zvětšování tabulky zabezpečení. Důsledkem je, že oprávnění k zobrazení připojeného obrázku vyprší za 15 minut.

Zobrazování obrázku je prováděno pomocí souboru `file.php`, který je umístěn v adresáři modulu. Kód zobrazující obrázek (proměnná `$id_obrazku` obsahuje hodnotu primárního klíče zvoleného obrázku z tabulky obrázků):

```
$secure_id = md5(session_id());
echo "<img src=\"".$_SESSION["HOME_URL"]
      . "moduly/clanky/file.php?obrazek=".$_id_obrazku
      . "&secure_id=".$_secure_id.\"\" alt=\"\" />";
```

Soubor `file.php`, na který se odkazujeme jako na zdroj tagu `img`, vyhodnocuje, zda je možné zobrazit obrázek:

```
$dotaz = "SELECT `id` FROM `".TAB_CLANKY_ZABEZPECENI
        . "` WHERE `id`='".addslashes($_GET['secure_id'])."' "
        . "AND `o_soubor`='".$_GET['obrazek']."' LIMIT 1";
$zabezpeceni = $db->sql_dotaz($dotaz);
if(sizeof($zabezpeceni) == 1){
    $dotaz = "SELECT `typ`, `soubor` FROM `".TAB_CLANKY_OBRAZKY
            . "` WHERE `id`='".$_GET['obrazek']."' LIMIT 1";
    $vysledek = $db->sql_dotaz($dotaz);
    if(sizeof($vysledek) > 0){
        header("Content-Type: " . $vysledek[0][0]);
        echo $vysledek[0][1];
    }
}else
    echo "Přístup odepřen -
        . "vaše oprávnění je po 15 minutách zrušeno.<br />"
        . "Zkuste <b>Aktualizovat</b> stránku portálu (ne tuto) "
        . "a otevřte soubor znovu.";
```

Výhoda této metody spočívá v tom, že i v případě přímého zobrazení skriptu `file.php` je otestováno oprávnění k zobrazení obrázku. Přímým zobrazením je například `http://server/moduly/clanky/file.php?obrazek=4&secure_id=168fc26a4c183cdeb06d39d947ecf572`.

Uvedené postupy umožňují řídit přístup k připojenému obrázku se stejnými omezeními práv jako pro objekt samotný (v našem příkladu je tedy zobrazení článku i obrázku k článku připojenému vázáno na oprávnění v kurzu a na to, zda je či není článek uveřejněn).

3.1.5 Čištění databáze

Při smazání kurzu jsou mazány i všechny podřízené kurzy. Uživatel, tedy může velmi snadno odstranit velkou část obsahu portálu, aniž by si plně uvědomil důsledky svého jednání. Jako ochrana před takovou neuvážeností slouží to, že při smazání kurzu se nevymaže obsah z modulů, které kurzu přináleží. Díky tomu je možné i po smazání kurzu obnovit jeho obsah.

Uvedená výhoda je však současně nevýhodou, protože dochází k zanášení databáze obsahem, který není studentům přístupný (protože leží ve smazaných kurzech). Proto je v každém modulu implementováno čištění databáze, které odstraňuje objekty patřící do smazaných kurzů.

Práva

Vyčištění databáze modulu může provést pouze hlavní uživatel, který má v kořenovém kurzu přístupný odkaz *Vyčistit databázi*, pomocí kterého vyčistí databázi aktivního modulu.

Kód

Samotné čištění databáze provádí metoda `clean_db` (soubor `moduly/clanky/class_clanky.php`) třídy modulu. Ta dále volá metodu pro mazání objektu (v tomto příkladu je volána metoda `del_clanek` – stejná třída i soubor), která vymaže objekt (článek) s hodnotou primárního klíče `$radek[0]`:

```
function clean_db($tabulka_temat){
    $dotaz = "SELECT `".TAB_CLANKY."`.`id`, `".$tabulka_temat."`.`id` "
        ."FROM `".
        .TAB_CLANKY."` LEFT JOIN `".$tabulka_temat."` ON `".
        .TAB_CLANKY."`.`tema`=`".$tabulka_temat."`.`id`";
    $vysledek = $this->sql_dotaz($dotaz);
    if(sizeof($vysledek) > 0)
        foreach($vysledek as $radek)
            if(empty($radek[1]))
                $this->del_clanek("", $radek[0], "1");
}
```

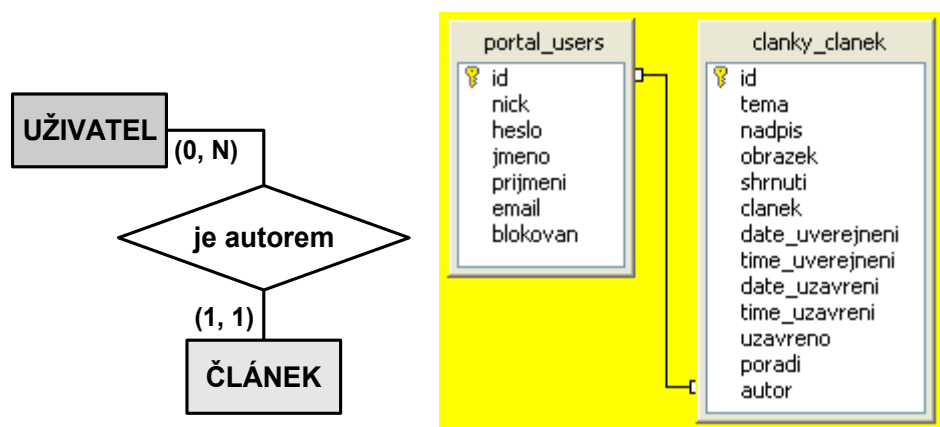
3.1.6 Autor objektu

U většiny modulů je evidován autor objektu. Příklad ukládání této informace do databáze je na obrázku 50 – jde o příklad z modulu *Články*.

Tato informace je důležitá zejména z hlediska zabezpečení přístupu k objektům. Konkrétně uživatel s rolí *příspěvateľ* může v modulech editovat a mazat pouze ty objekty, které sám vytvořil.

V testovacím modulu je toto autorství navíc důležité ke sledování toho, komu patří příslušný výsledek vypracovaného testu a samozřejmě také k tomu, aby se studenti mohli vracet pouze k těm výsledkům testů, které v minulosti vypracovali sami.

Důležitost znalosti autora je podobná i v modulu *Diskuse* a ostatních modulech.



Obrázek 50: Příklad uložení autora objektu v databázi (pro modul *Články*)

Při zobrazení objektu je často zobrazena přezdívka autora, která je hypertextovým odkazem vedoucím na podrobnosti o tomto uživateli.

3.2 Modul *Články*

Prvním modulem, o kterém se zmíníme, je modul *Články*. Modul tohoto typu nesmí chybět v žádném portálu. Pokud bychom chtěli e-learningový portál nasadit v roli redakčního webu, mohl by tento modul být jediným instalovaným modulem.

Modul *Články* obsahuje všechny společné vlastnosti modulů uvedené v kapitole 3.1 (proto nám také sloužil jako příklad fungování těchto vlastností).

Základním požadavkem na tento modul je možnost zobrazit články v přehledném seznamu, ze kterého si student vybere ten článek, jehož podrobnosti jej zajímají. Dále možnost vložit obrázek je v tomto modulu nutností, protože výklad článku může být popisem grafického materiálu a nebo naopak grafický materiál může doplňovat text článku.

Z funkčního hlediska je rovněž potřeba povolit uživateli základní formátování textu nejlépe pomocí vkládání HTML tagů, které jsou obecně známé.

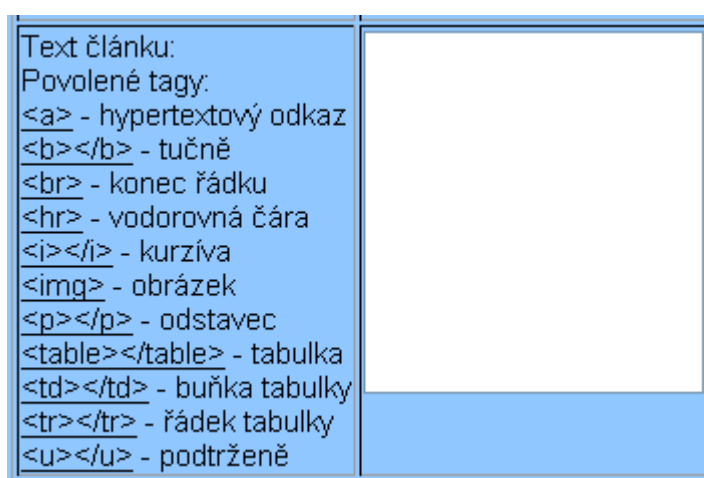
Značky HTML

Povolit, či nepovolit značky HTML? V e-learningovém portálu byla zvolena varianta povolit značky HTML, což je samozřejmě z hlediska bezpečnosti portálu velmi citlivé místo. Proto je vkládání značek kontrolováno a omezeno na nejnutnější. Navíc validita portálu je závislá na validitě jednotlivých příspěvků (viz kapitola 1.4 *WWW, HTML, CSS a validita* na straně 20). Výhodou této volby je bezesporu to, že nenutí uživatele, aby se učil nějaký nový systém značek (například oblíbené `[b]...[/b]` namísto `...` atp.).

Ve výchozím stavu jsou všechny html značky zakázané. Povolení značky se provádí v konfiguračním souboru `include/config.php` modulu *Články* přidáním značky do pole:

```
$CLANKY_TAGY = array(  
    array("hr", 0, "vodorovná čára"),  
    array("p", 1, "odstavec"),  
    array("b", 1, "tučně"),  
    array("i", 1, "kurzíva")  
);
```

Například povolení značky `...`: Vytvoří se tříprvkové pole. První prvek představuje samotnou značku; druhý vyjadřuje, zda je značka párová; třetí obsahuje vysvětlující popis značky, který se zobrazí v seznamu povolených tagů na stránce při vytváření článku. Zmíněný seznam současně slouží jako panel pro vkládání tagů.



Obrázek 51: Seznam povolených HTML značek

3.2.1 Přístupová práva

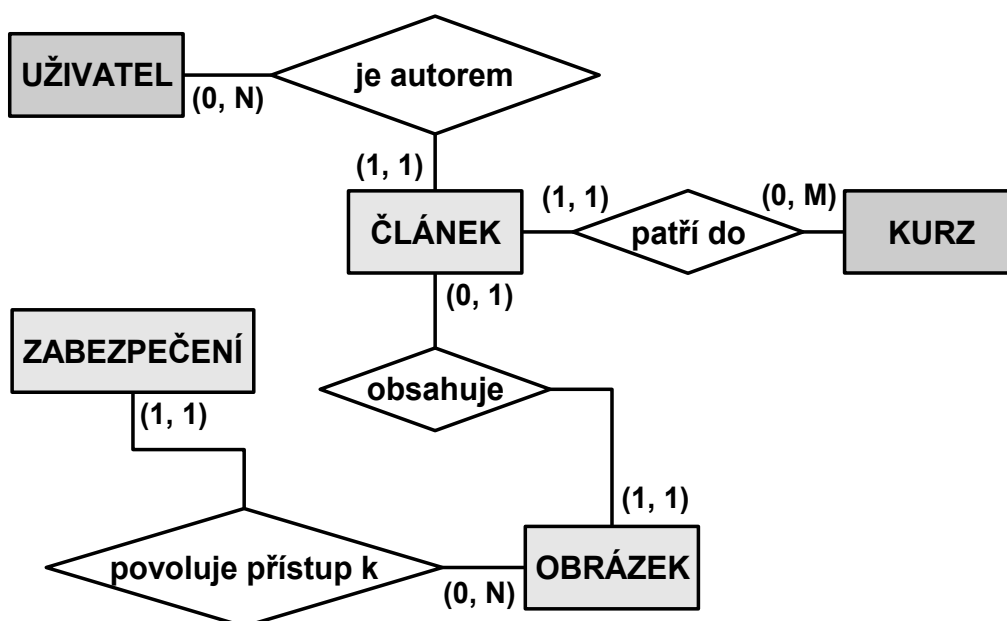
Při práci v modulu *Články* mají jednotliví uživatelé různá oprávnění. Vychází se z role, kterou uživatel získal při ověřování svého přístupu do vybraného kurzu. Akce, které mohou uživatelé s různými rolemi vykonávat, jsou shrnuty v tabulce 3.

	<i>Vlastník a redaktor</i>	<i>Příspěvateľ</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Vytvoření článku	X	X			
Změna pořadí článků	X				
Editace a výmaz jakéhokoli článku	X				
Editace a výmaz vlastního článku	X	X			
Čtení článku	X	X	X	X	

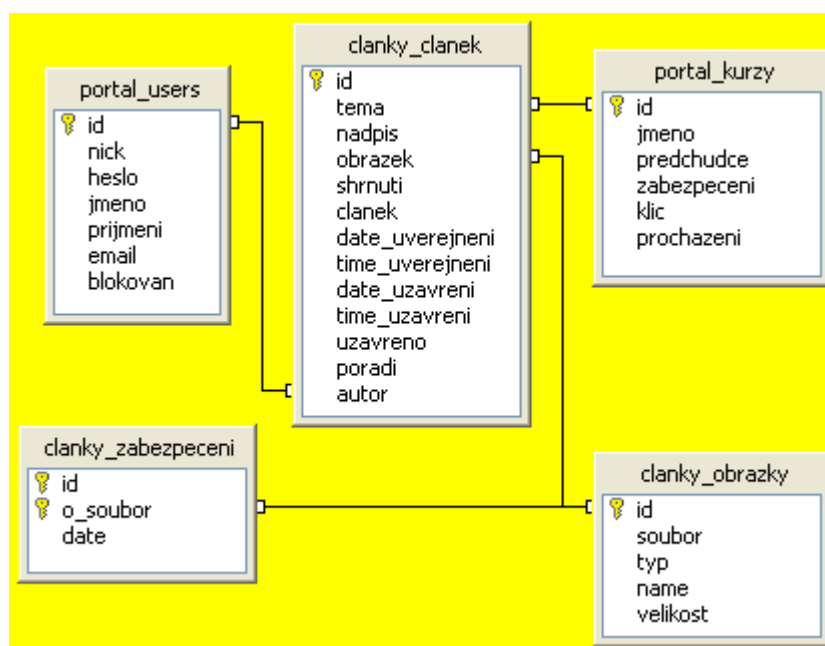
Tabulka 3: Oprávnění v modulu *Články*

3.2.2 Databáze

Na základě konvencí rozebraných v kapitole 1.2.3 nyní zakreslíme ER diagram části databáze, která funkčně přináleží modulu *Články* a znázornění tabulek, atributů a relací.



Obrázek 52: ER diagram databáze modulu *Články*



Obrázek 53: Znáznornění tabulek, atributů a relací mezi tabulkami

Význam entit *OBRÁZEK* a *ZABEZPEČENÍ* již byl vysvětlen v kapitole 3.1.4 *Připojené obrázky* na straně 54.

Každé entitě *ČLÁNEK* je přiřazena entita *UŽIVATEL*, který uživatel je autorem článku. Dále musí být každé entitě *ČLÁNEK* přiřazena entita *KURZ* – do kterého kurzu článek patří.

3.3 Modul Soubory

Aby mohl učitel předávat studentům materiál prakticky jakéhokoli typu, je součástí portálu i modul *Soubory*. Pomocí tohoto modulu je možné nahrát do databáze portálu (a tedy i zabezpečit) jakýkoli typ souboru: Od dokumentů ve formátu .pdf, přes různé .doc, .xls, .odt, ... až po multimediální soubory obsahující ukázky hudby nebo flash animace.

V modulu *Soubory* jsou ze společných vlastností zakomponovány v podstatě všechny, ovšem možnost připojování obrázků je modifikována.

Podobně jako v modulu *Články*, se nejprve zobrazí seznam souborů zařazených do kurzu. Pokud je některý z nich vybrán, zobrazí se komentář k souboru a informace o souboru (typ, název a velikost). Díky tomuto zobrazování „nadvakrát“ je možné pracovat s URL konkrétního souboru.

vyroubat: **Co to vlastně dělá 2**

Zde je soubor ve formátu pdf. Tento formát bych doporučil jako formát na výměnu dokumentů spíše, než použít .doc, .ppt, ... V tomto souboru najdete krátkou prezentaci vytvořenou v programu OpenOffice.org Impress.

Typ: application/pdf, **Jméno:** prezentace.pdf, **Velikost:** 217.42 kB

[Zobrazit](#) - [Uložit](#)

Obrázek 54: Informace k vybranému souboru

Samotná práce se souborem se provádí pomocí odkazů *Zobrazit* (pokud je typ souboru asociován s příslušným programem, je v něm přímo otevřen) a *Uložit* (uložení souboru). Přesné chování samozřejmě záleží na klientském programu – prohlížeči, kterým se ke stránkám přistupuje. Technicky je rozdíl v hlavičce, která je prohlížeči odeslána (ukázka kódu je ze souboru `file.php` modulu *Soubory*).

```
switch($_GET['soubory_akce']){
    case "show" : //Zobrazit soubor
        header("Content-Type: ".$vysledek[0][0]);
        break;
    case "save" : //Uložit soubor
        header("Content-Description: File Transfer");
        header("Content-Type: application/force-download");
        header("Content-Disposition: attachment; filename=\"\"
            . $vysledek[0][2].\"\"");
        break;
    default : //Jinak
        die("Neoprávněný přístup");
}
echo $vysledek[0][1];
```

Pole `$vysledek[0]` je tvořeno informacemi o připojeném souboru. Proměnná `$vysledek[0][1]` představuje samotný soubor.

Zabezpečení souboru je řešeno podobným způsobem jaký byl popsán v kapitole 3.1.4 *Připojené obrázky* na straně 54 v části *Zabezpečení připojeného obrázku* na straně 56.

Flashové animace

Oblíbeným multimediálním prvkem při e-learningové výuce jsou flashové animace. Pomocí těchto animací je možné vytvořit jednoduchá pohyblivá schémata, interaktivní výukové prezentace, testy, integrovat video atd. Pro vytvoření flashových animací je možné použít například nástroj Macromedia Flash nebo mnohem jednodušší Swish. Takto vytvořenou animaci je možné nahrát do modulu *Soubory* – v podstatě jde o soubor s příponou `.swf`.

3.3.1 Přístupová práva

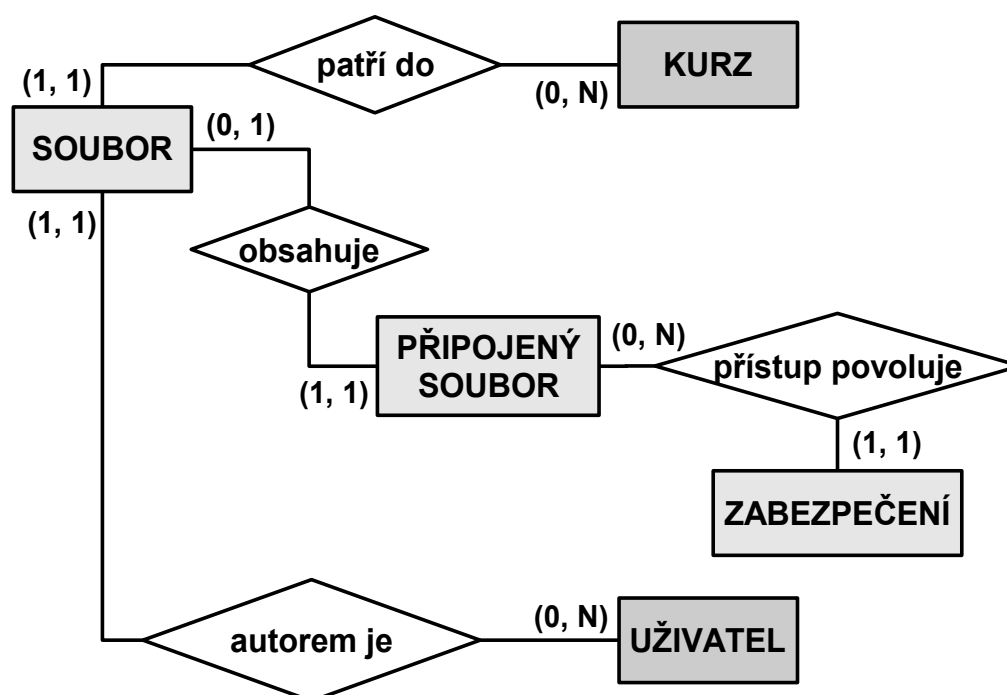
Na základě role, kterou uživatel získá při ověřování přístupu ke kurzu, jsou uživateli povolovány určité aktivity v modulu *Soubory* – jejich rozbor je uveden v tabulce 4.

	<i>Vlastník a redaktor</i>	<i>Příspěvateľ</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Nahrání souboru	X	X			
Změna pořadí souborů	X				
Editace a výmaz jakéhokoli souboru	X				
Editace a výmaz vlastního souboru	X	X			
Prohlížení a download souboru	X	X	X	X	

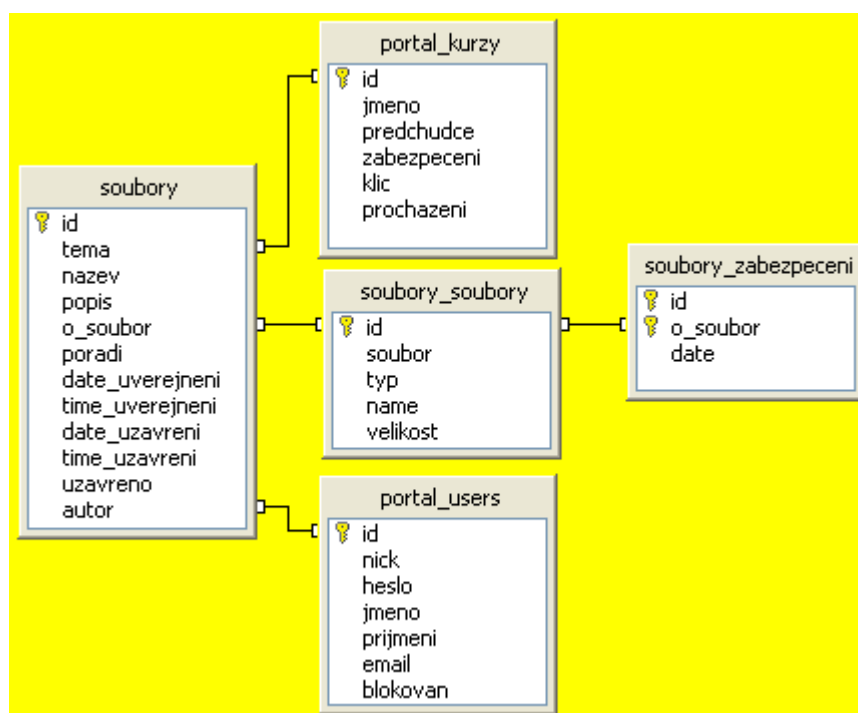
Tabulka 4: Oprávnění v modulu *Soubory*

3.3.2 Databáze

Na základě konvencí rozebraných v kapitole 1.2.3 nyní zakreslíme ER diagram databáze modulu *Soubory* a znázornění tabulek, atributů a relací.



Obrázek 55: ER diagram databáze modulu *Soubory*



Obrázek 56: Znáznornění tabulek, atributů a relací mezi tabulkami

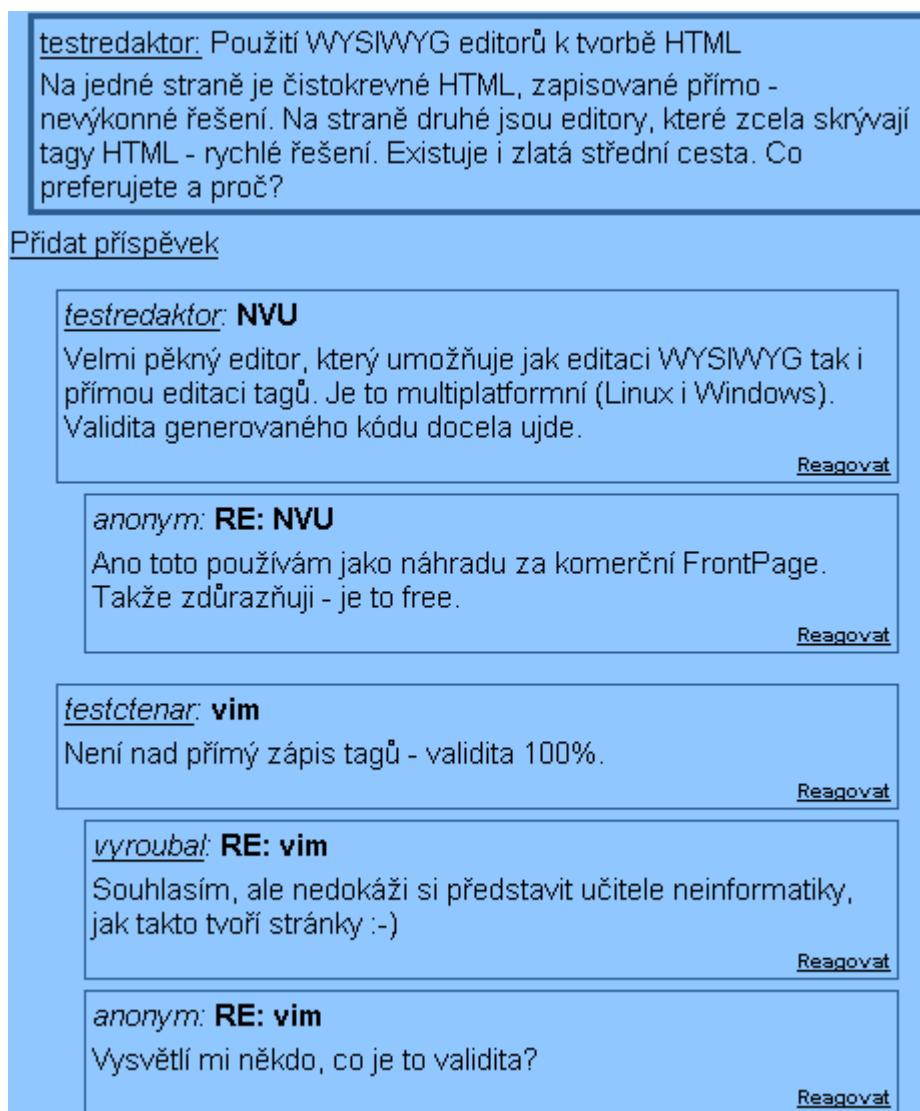
Entitní typ *SOUBOR* představuje informace, které soubor popisují – nadpis (atribut *nazev*), popis – a informace týkající se autora, kurzu, do kterého soubor patří, a uveřejnění (resp. Stažení) souboru. Entitní typ *PŘIPOJENÝ SOUBOR* (tabulka *soubory_soubory*) představuje soubory samotné (například *prezentace.pdf*) uložené v databázi.

3.4 Modul Diskuse

Diskuse je prvním modulem, který umožňuje interakci mezi učiteli a studenty, mezi jednotlivými studenty i mezi jednotlivými učiteli. Pro učitele a rovněž pro studenta poskytuje tento modul informace zpětné vazby a pomáhá ke zvýšení míry pochopení učiva, což jsou jeho základní přínosy. Bez tohoto modulu by e-learningový portál nemohl být úplný, protože informační tok by byl jednosměrný od učitele ke studentům a informace zpětné vazby by byly poskytovány až v okamžiku, kdy je student testován – neboli většinou příliš pozdě.

Ze společných vlastností modulů rozebíraných v kapitole 3.1 *Společné vlastnosti modulů* není v modulu *Diskuse* implementována možnost připojování obrázků.

Tento modul nejprve zobrazuje seznam všech diskusí v kurzu a při výběru konkrétní diskuse se zobrazí téma diskuse i s připojenými příspěvky:



Obrázek 57: Zobrazení diskuse

Příspěvky v diskusi jsou řazeny do stromové struktury. Počet úrovní reakcí není omezen. Avšak při zobrazování příspěvků se provede odsazení zleva pouze pro prvních osm úrovní vnoření. Devátá a každá další úroveň mají stejné odsazení jako úroveň osmá. Vzhled jednotlivých úrovní je řízen pomocí kaskádových stylů a vrstev, jak ukazuje následující příklad, zobrazení stromu příspěvků v diskusi (jde pouze o začátek kódu, který je součástí metody `zobraz_strom` objektu `diskuse` v souboru `class_diskuse.php` modulu *Diskuse*):

```
for($j = 0; $j < $sloupcu; $j++){
    if($strom[$i][$j]){
        echo "<div class=\"diskuse_prispevek_uroven_\"";
        if($j < 7) echo $j;
        else echo 7;
        echo ">";
        ... }
    }
```

U příspěvků se zobrazuje přezdívka autora příspěvku – jde o odkaz, který vede na stránku s informacemi o autorovi.

Uzamčení diskuse

Kromě načasování uveřejnění a uzavření diskuse (viz kapitola 3.1.3 *Načasování uveřejnění a stažení*, str. 52) je možné diskusi uzamknout. Pokud je diskuse uzamčena, mohou sice uživatelé zobrazit její obsah (příspěvky), ale nemohou přidávat nové příspěvky a reakce.

Toto nastavení diskuse je praktické například v situaci, kdy je diskutovaná otázka dostatečně probrána a vyřešena. Uzamknutím diskuse se zabrání dalšímu, zbytečnému rozměňování tématu a tudíž i znepráhlednění diskuse. Ale díky tomu, že diskuse je nadále viditelná, mohou si uživatelé přečíst výsledek diskuse.

Oprávnění k uzamčení diskuse je totožné s oprávněním k editaci vlastností diskuse. Role s oprávněním k této změně budou zmíněny dále.

Editace a smazání příspěvku

Určení uživatelé mohou editovat nebo odstranit příspěvek z diskuse. Tato možnost sice připomíná metody cenzury, ale mluvíme o diskusi, která je součástí e-learningového portálu, ve kterém musí mít učitel možnost výchovného působení – podobně jako při výuce ve třídě.

Při vymazání příspěvku je ovšem nutné vědět, že spolu s příspěvkem budou vymazány všechny reakce (přímé i nepřímé). Neboli terminologií grafů: Spolu s příspěvkem bude odstraněn i celý podstrom příspěvků, který je mu podřízen.

3.4.1 Přístupová práva

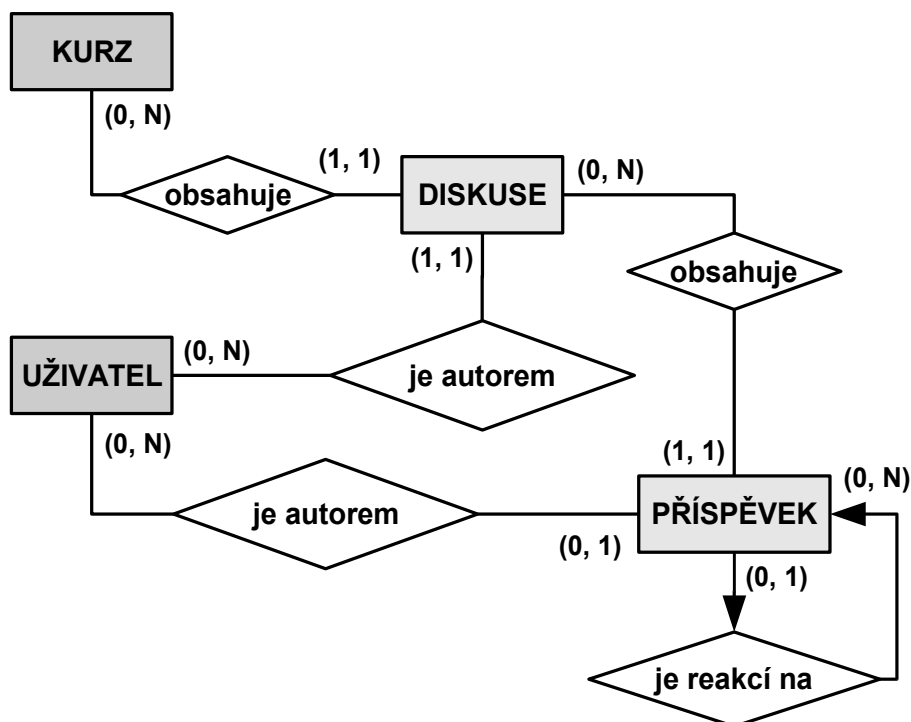
Oprávnění navázaná na uživatelské role v modulu *Diskuse* shrnuje následující tabulka 5.

	<i>Vlastník a redaktor</i>	<i>Příspěvatel</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Vytvoření diskuse	X	X			
Změna pořadí diskusí	X				
Editace a výmaz jakékoli diskuse	X				
Editace a výmaz vlastní diskuse	X	X			
Zobrazení diskuse	X	X	X	X	
Přidání příspěvku do diskuse	X	X	X	X	
Editace a výmaz příspěvku v jakékoli diskusi	X				
Editace a výmaz příspěvku ve vlastní diskusi	X	X			

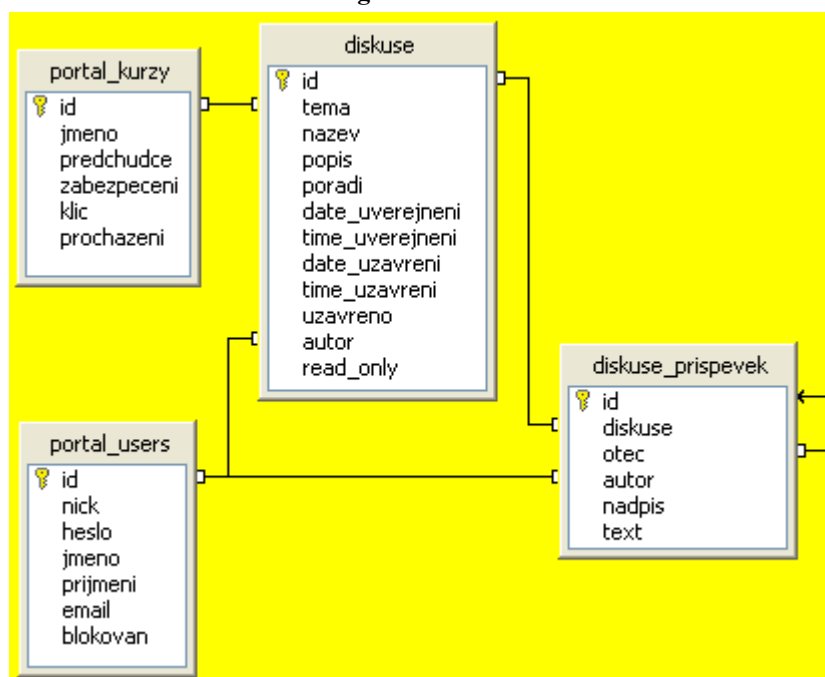
Tabulka 5: Oprávnění v modulu *Diskuse*

3.4.2 Databáze

Na základě konvencí rozebraných v kapitole 1.2.3 nyní zakreslíme ER diagram databáze modulu *Soubory* a znázornění tabulek, atributů a relací.



Obrázek 58: ER diagram databáze modulu *Diskuse*



Obrázek 59: Znázornění tabulek, atributů a relací mezi tabulkami

Blíže se pozastavíme nad entitním typem *PŘÍSPĚVEK*. Tento entitní typ je v relaci s entitním typem *UŽIVATEL* – jde o autorství příspěvku v diskusi. Členství entity

PRÍSPĚVEK v této relaci je typu (0, 1), protože autorem příspěvku může být anonymní uživatel – takový příspěvek tedy autora v podstatě nemá.

Entita *PRÍSPĚVEK* je rovněž v rekurentní relaci sama se sebou. Ve směru od podřízených příspěvků k nadřazeným je relace typu (0, 1), protože některé příspěvky nejsou reakcemi na příspěvky, ale navazují přímo na téma diskuse (neboli příspěvky první úrovně); v opačném směru je relace typu (0, N), protože na každý příspěvek může být navázán libovolný počet reakcí (i žádná).

3.5 Modul Testy

Testovací modul e-learningového portálu je z pohledu programátora bezesporu nejnáročnějším modulem. Z pohledu učitele však jde o velmi užitečný nástroj. Pomocí modulu *Testy* je možné provádět celou škálu testování od cvičných testů, které budou skládat studenti sami pro sebekontrolu a které se budou rovnou vyhodnocovat, až po klasifikované testování, kde může učitel k odpovědím na jednotlivé otázky doplnit slovní komentář a vlastní systém hodnocení.

Testy svádějí studenty k tomu, aby za každou cenu (i v situaci, kdy vědí, že neznají správnou odpověď) zvolili u uzavřených otázek některou z možností. Aby se toto nežádoucí chování eliminovalo, je v testovacím modulu portálu začleněn nejen systém bodů za správné odpovědi, ale i systém penalizací v případě odpovědí špatných.

Studenti se mohou k absolvovaným testům vracet a sledovat svůj pokrok při studiu předkládané látky. Rovněž učitel si může procházet výsledky testů a dále s nimi pracovat podobně jako při zpracovávání běžných testů.

Testovací modul poskytuje velmi cenné informace zpětné vazby pro žáka i pro učitele. Modul *Testy* tedy zaujímá klíčovou úlohu v edukačních funkcích e-learningového portálu.

Struktura dat a logika práce s modulem *Testy* se dá rozdělit do tří vzájemně souvisejících celků: Otázky, testy a výsledky testů. Pojednáme tedy o každé z těchto tří částí samostatně, přičemž nalinkujeme vzájemné vazby. V následujícím rozboru se budeme hlouběji věnovat těm skutečnostem, které jsou klíčové pro vytvoření nového typu otázky.

3.5.1 Otázky

Základními stavebními kameny každého testu jsou otázky. Vytvořené otázky jsou součástí kurzu a je možné je používat pro vytváření testů v daném kurzu.

Při vytváření otázky se vychází ze čtyř základních typů otázek. Úvodní část definice otázky je společná všem typům otázek a následně se pracuje s částí specifickou pro daný typ otázky.

Společné vlastnosti všech typů otázek

V obecné části otázky se při jejím vytváření definuje *název otázky*, pod kterým bude figurovat v seznamu otázek v kurzu.

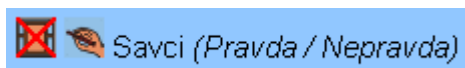
Body představují hodnotu, která je studentovi připočtena při správné odpovědi, a *penalizace* hodnotu, která je naopak odečtena při odpovědi špatné.

Ke každé otázce je možné připojit obrázek. Funkčnost a práce s takto připojeným obrázkem byla popsána v kapitole 3.1.4 *Připojené obrázky* na straně 54.

Název otázky:	<input type="text" value="Savci"/>
Penalizace:	<input type="text" value="0.5"/>
Body:	<input type="text" value="3.5"/>
Obrázek k otázce	<input type="text"/> Procházet...

Obrázek 60: Vytvoření otázky - společná část

Pokud má uživatel v kurzu roli s oprávněním k editaci či výmazu otázky, zobrazí se mu v seznamu otázek ikony pro příslušné akce (v závorce je u otázky připsaný typ otázky):



Obrázek 61: Otázka s administračními ikonami

Nyní probereme jednotlivé typy otázek. Pro všechny je při vytváření shodná část zobrazená na obrázku 60.

Typ otázky *Pravda / Nepravda*

Nejjednodušším a nejzákladnějším typem otázky je výběr, zda uvedené tvrzení je nebo není pravdivé. Při vytváření nové otázky tohoto typu se zadává tvrzení, a zda je správnou odpovědí *pravda* nebo *nepravda*.

Zadání:	<input type="text" value="Na obrázku je savec."/>
Správná odpověď:	<input type="text" value="Nepravda"/> ▼

Obrázek 62: Nová otázka typu *Pravda / Nepravda*

Typ otázky *Výběr možností*

Druhým obvyklým uzavřeným typem otázky je výběr správné možnosti. V e-learningovém portálu je umožněno, aby z nabízených možností bylo správných několik (i všechny), jedna, nebo i žádná.

Maximální počet možností, které je možné připojit k otázce, je přednastaven na 10. Samozřejmě se dá velmi snadno změnit a to úpravou hodnoty konstanty `TESTY_MOZNOSTI_PO CET_MOZNOSTI` v souboru `config.php` v adresáři modulu *Testy*. (Zde uvedená hodnota 9, znamená, že bude 10 možností, číselovaných 0 až 9.)

```
| define("TESTY_MOZNOSTI_PO CET_MOZNOSTI", '9');
```

Není nutné využít všechny možnosti. Při ukládání otázky do databáze nedojde k uložení vynechaných polí pro možnosti (neboli prázdné možnosti nebudou v databázi zabírat zbytečně místo). Mají-li tedy být u otázky čtyři možnosti, je nutné vyplnit 1. až 4. možnost a 5. a další nechat prázdné.

Zadání:	Člověk patří do	
1. možnost:	řádu Nehetnatci	<input checked="" type="checkbox"/> Správná odpověď
2. možnost:	podřádu Vyšší primáti	<input type="checkbox"/> Správná odpověď
3. možnost:	nadčeledi Lidé	<input type="checkbox"/> Správná odpověď
4. možnost:	rodu Člověk	<input checked="" type="checkbox"/> Správná odpověď
5. možnost:		<input type="checkbox"/> Správná odpověď

Obrázek 63: Nová otázka typu *Výběr možností*

Typ otázky *Slovní odpověď*

Otázka typu slovní odpověď se dá použít dvojím způsobem. Prvním případem je, když je odpověď jednoduchá (nejlépe jednoslovná). Ta se při automatickém vyhodnocování testu porovná se vzorem vloženým při vytváření otázky. Pokud jsou odpověď a vzor shodné, vyhodnotí se odpověď jako správná. Vzhledem k tomu, že jde o přímý zápis odpovědi studenta, může v tomto případě dojít k tomu, že odpověď, která je v podstatě správná, ale zapsaná jinak než vzor, bude vyhodnocena jako nesprávná. Aby k takovým situacím nedocházelo, jsou k tomuto typu otázky přidány dvě funkce pro rozpoznání „podobné odpovědi“.

Při vytváření otázky je možné zvolit, zda má portál při vyhodnocování *rozlišovat velikost písmen* (neboli zda má být casesensitive). Při samotném vyhodnocování slovní odpovědi jsou vynechány všechny netisknutelné znaky na začátku a na konci řetězce odpovědi (např. mezery) pomocí PHP funkce `trim()`. Díky těmto dvěma mechanismům je zmírněna ošemetnost slovní odpovědi na otázku.

Druhý způsob, jak používat typ otázky slovní odpověď, je bez použití automatického vyhodnocování. Oprava, komentář a bodování jsou doplněny učitelem, který zváží správnost řešení testu. Tato varianta použití je možná a testovací modul e-learningového portálu se posouvá do polohy správce „odevzdaných“ testů. Nicméně v některých situacích je tento způsob použití velmi praktický a šetří učiteli práci tím, že testy jsou uchovávány v informačním systému.

Zadání:	Savci jsou latinsky:
Správný výsledek:	Mammalia
Velikost písmen:	<input type="checkbox"/> Rozlišovat velikost písmen

Obrázek 64: Nová otázka typu *Slovní odpověď*

Typ otázky *Vypočtená úloha*

Pro učitele většiny přírodovědných disciplín je nutné zadávat studentům příklady k vypočítání. Díky zaokrouhlovacím chybám je však třeba při porovnávání studentova výsledku pracovat s chybou (odchylkou). Otázky typu *Vypočtená úloha* umí pracovat s chybou (respektive *Tolerance*), kterou zadá učitel při tvorbě otázky.

Zadání:	Vypočtete obsah kruhu
Správný výsledek:	10.05
Tolerance:	±0.01

Obrázek 65: Nová otázka typu *Vypočtená úloha*

3.5.1.1 Přístupová práva

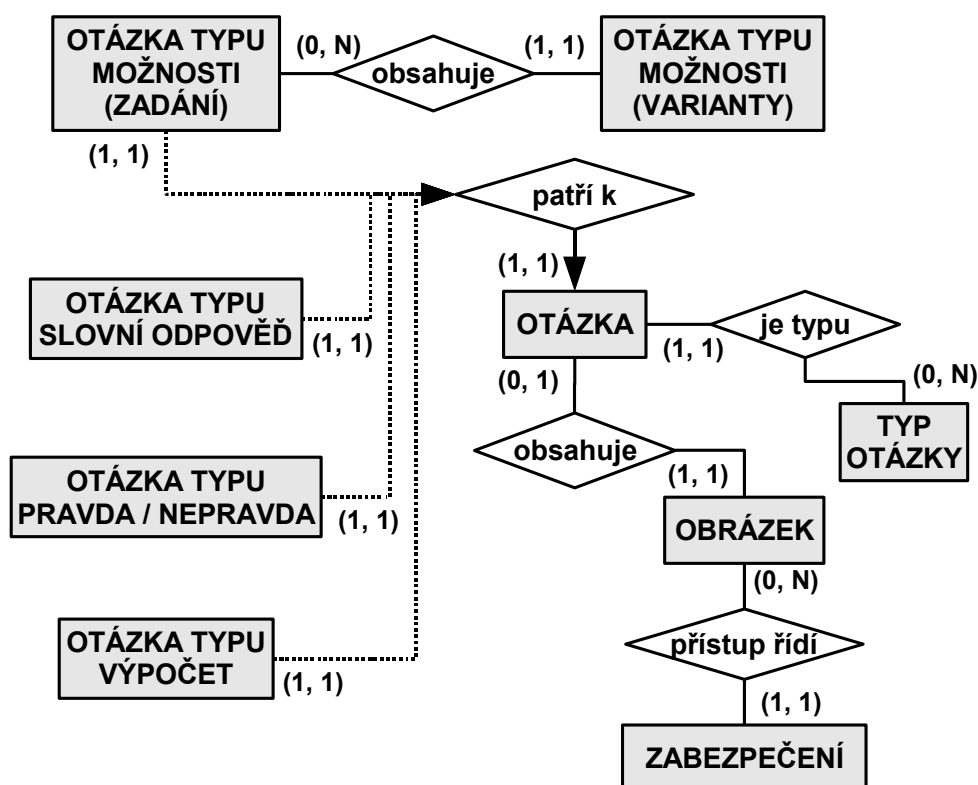
Role a jejich oprávnění při práci s otázkami v modulu *Testy* jsou shrnuty v tabulce 6. Ze zřejmých důvodů jsou oprávnění přidělena pouze rolím *vlastník*, *redaktor* a *příspěvatel*.

	<i>Vlastník a redaktor</i>	<i>Příspěvatel</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Vytvoření otázky	X	X			
Editace a výmaz jakékoli otázky	X				
Editace a výmaz vlastní otázky	X	X			

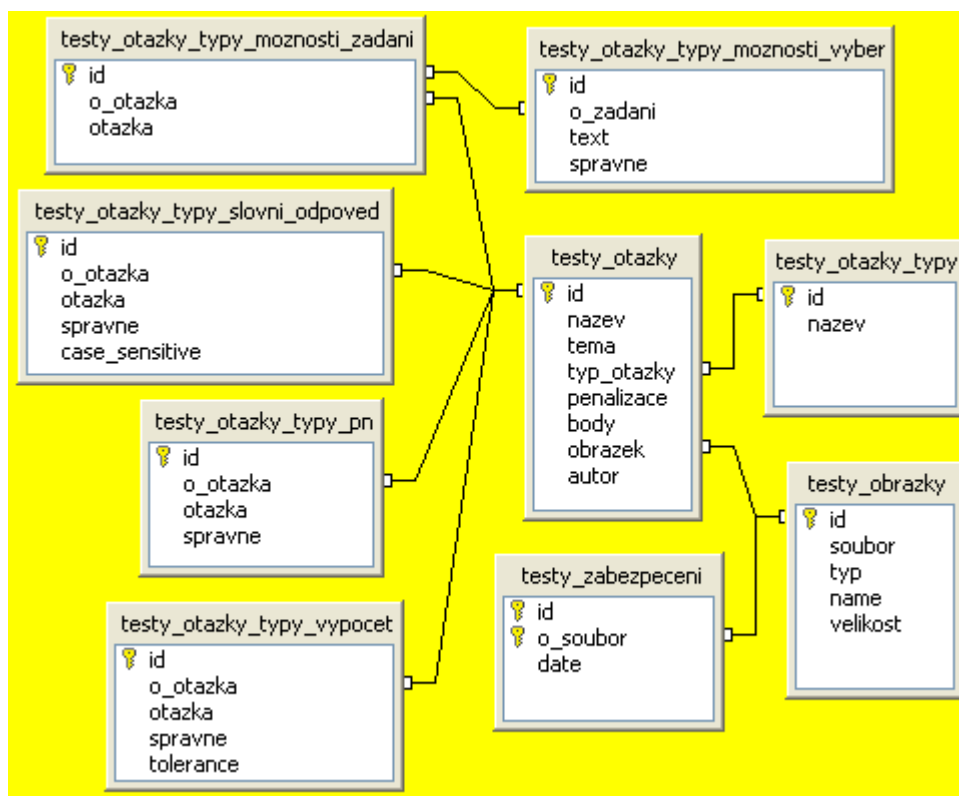
Tabulka 6: Oprávnění v modulu *Testy* – otázky

3.5.1.2 Databáze

Schémata části databáze, která uchovává informace o otázkách, jsou na obrázcích 66 a 67. Tabulky s názvem s prefixem *testy_otazky_tpy...* (entitní typ *OTÁZKA TYPU ...*) náleží jednotlivým typům otázek. Seznam typů otázek je k dispozici v tabulce *testy_otazky_tpy* (entitní typ *TYP OTÁZKY*). Tabulka *testy_otazky* (entitní typ *OTÁZKA*) obsahuje obecné informace o otázce, které jsou společné pro všechny typy otázek.



Obrázek 66: ER diagram databáze modulu *Testy* - otázky



Obrázek 67: Znáznornění tabulek, atributů a relací mezi tabulkami

Při práci s otázkou se nejprve načtou obecné informace z tabulky *testy_otazky* a poté je podle hodnoty atributu *typ_otazky* rozhodnuto, která z tabulek *testy_otazky_typy_...* se použije pro načtení dalších informací specifických pro daný typ otázky.

Data

Od začátku jsou v databázi modulu *Testy* následující data:

- tabulka *testy_otazky_typy* – jsou zde zaznamenány názvy typů otázek (bude dále vysvětleno)

3.5.1.3 Vytvoření nového typu otázky

Pro vytvoření typu otázky je nutná alespoň základní znalost PHP. Testovací modul je však sestaven tak, aby nebylo třeba při přidávání typu otázky velkých zásahů do kódu. Postup vytvoření si ukážeme na nejjednodušším příkladu – na typu otázky *Pravda / Nepravda*.

Databáze pro nový typ otázky

Nejprve je nutné vytvořit tabulku (či tabulky), které budou v databázi uchovávat data o části otázky specifické pro vytvářený typ. V našem příkladu jde o tabulku *testy_otazky_typy_pn* (viz obrázky 66 a 67).

Jakmile jsou tabulky vytvořeny, zaznamenají se jejich názvy do konstant definovaných v souboru `config.php` v adresáři modulu, aby odkazy v kódu neukazovaly pevně na název tabulky, ale používaly tuto konstantu:

```
| define("TAB_TESTY_OTAZKY_TYP_PN", TAB_PREFIX.'testy_otazky_typy_pn');
```

Zaregistrování typu otázky

Typ otázky se registruje na dvou místech.

Nejprve je nutné přidat záznam do tabulky *testy_otazky_typy*, kde atribut *název* obsahuje jako hodnotu pojmenování typu otázky – toto pojmenování se bude používat ve všech zobrazeních portálu.

Poté se nadefinuje konstanta v souboru `config.php` v adresáři modulu, jejíž hodnota odpovídá primárnímu klíči z tabulky *testy_otazky_typy*.

```
| define("TESTY_PRAVDA_NEPRAVDA", '1');
```

Metoda typu otázky

Dále je nutné vytvořit novou metodu ve třídě *otazka* (soubor `class_otazky.php`). V našem příkladu jde o metodu *otazka_pravda_nepravda*.

Tato metoda je v podstatě přepínačem `switch`, který obsluhuje jednotlivé události (řídící proměnnou přepínače je `$akce`, která je předávána metodě při jejím volání):

- **form_new** – formulář pro vytvoření nové otázky daného typu.
Nejde o celý formulář, ale pouze o ty formulářové prvky, které jsou specifické pro daný typ otázky.
- **db_insert** – uložení nové otázky do databáze.
Jde opět pouze o ta data, která jsou specifická pro daný typ. Ukládá se do tabulky, kterou jsme si vytvořili v prvním kroku (*Databáze pro nový typ otázky* na str. 75).
- **form_edit** – formulářové prvky pro editaci otázky.
Která otázka bude editována je určeno hodnotou parametru `$id_otazky`.
- **db_update** – uložení (update) změn provedených na otázce do databáze.
Která otázka bude updatována je určeno hodnotou parametru `$id_otazky`.
- **form_test** – zobrazení otázky ve chvíli, kdy student pracuje na testu.
Která otázka bude zobrazena je určeno hodnotou parametru `$id_otazky`.
- **form_after_test** – provede se po odeslání testu studentem.
Která otázka bude zpracována je určeno hodnotou parametru `$id_otazky`.
Tato větev má dvě základní funkce: Provádí vyhodnocení, zda byla

odpověď studenta správná, a počítá body a penalizaci. Rovněž sestavuje proměnnou `$retezec`, která se zobrazí, pokud je povoleno zobrazení vyhodnocení testu studentovi ihned po dokončení testu.

Parametry `$body_spravne` a `$penalizace` se předávají metodě podle nastavení obecné části otázky.

V této větvi vrací metoda pole, které obsahuje počet dosažených bodů za otázku, řetězec, který se zobrazí je-li povoleno zobrazení automatického vyhodnocení a hodnotu představující odpověď studenta (jak bude ukázáno dále, tato hodnota se ukládá do tabulky *testy_odpoved_otazka*).

- **form_hodnoceni** – provede se, pokud si učitel nebo student zpětně prohlíží výsledky testů.

Která otázka bude zobrazena je určeno hodnotou parametru `$id_otazky`.

Zobrazí otázku, studentovu odpověď a automatické hodnocení, pokud je to povoleno (hodnota boolean parametru `$hodnoceni_automaticke`).

Parametry `$body_spravne` a `$penalizace` mají stejný význam jako ve větvi `form_after_test`.

Parametr `$odpoved` obsahuje studentovu odpověď načtenou z tabulky *testy_odpoved_otazka* a předanou metodě.

V této větvi metoda vrací pouze počet bodů získaných za danou otázku.

- **db_delete** – vymaže z databáze část otázky specifickou pro daný typ. Která otázka bude odstraněna je určeno hodnotou parametru `$id_otazky`.

Přidání metody typu otázky do hlavního přepínače otázky

Vždy, když je prováděna některá z akcí se specifickou částí otázky, je používána metoda `switch_typ_otazky` (třída `otazka` v souboru `class_otazky.php`). Opět jde o přepínač, který rozhodne, která metoda se má použít – rozhoduje se na základě toho, o jaký typ otázky jde. Zde je nutné vytvořit novou větev, pro nový typ otázky:

```
function switch_typ_otazky($akce, $typ_otazky, $id_otazky,
    $odpoved = 0, $body_spravne = 0,
    $penalizace = 0, $hodnoceni_automaticke = 1){
    switch($typ_otazky){
        case TESTY_PRAVDA_NEPRAVDA :
            return $this->otazka_pravda_nepravda($akce, $id_otazky, $odpoved,
                $body_spravne, $penalizace,
                $hodnoceni_automaticke);

            break;
        default :
            die("Neopravneny pristup");
    }
}
```

V této chvíli je možné používat nový typ otázky v e-learningovém portálu.

3.5.2 Testy

Otázky již máme prozkoumané a vytvořené. Přistupme tedy k práci s testy.

Vytvoření nového testu

Test je možné v kurzu sestavit s použitím jakékoli otázky patřící do kurzu (bez ohledu na to, kdo tuto otázku vytvořil). Při sestavování testu se nejprve nastaví parametry testu (viz Obrázek 69) a zvolí se testovací otázky. Ve druhém kroku se pak určí pořadí otázek v testu (viz Obrázek 70).

Parametry, které se pro test nastavují, jsou:

- **Název** – pod tímto názvem bude test figurovat v seznamu testů.
- **Vyhodnocení testu** – je-li tato funkce aktivována, zobrazí se studentovi okamžitě po dokončení testu vyhodnocení včetně správných odpovědí. Je nutné mít zapnuté *Automatické vyhodnocení* (bude dále vysvětleno). Je-li třída plná studentů a všichni současně skládají stejný test, není vhodné povolovat *Vyhodnocení testu*. V této konkrétní situaci je navíc ještě dobré zakázat možnost *Prohlížení výsledků* (bude dále vysvětleno).

- **Způsob vyhodnocení testu:**
 - **Automatické vyhodnocení** – vyhodnocování testu a výpočet dosažených bodů se provede automaticky na základě správných odpovědí zadaných při tvorbě otázky. Toto vyhodnocení je ukázáno na následujícím obrázku:

Savci 3 Savci jsou latinsky: <input type="text" value="mammalia"/>	správně Požadovaný výsledek: <input type="text" value="Mammalia"/> Nezáleží na velikosti písmen. Body: 1 (max: 1, penalizace: 0.1)
---	---

Obrázek 68: Automatické vyhodnocení otázky

- **Vyhodnocení učitelem** – neprovede se automatické vyhodnocení, ale učitel ručně doplní své vlastní hodnocení k testovým otázkám a celému testu.
 - **Vyhodnocení automatické i učitelem** – kombinace obou předchozích.
 - **Žádné vyhodnocení** – takto nastavený test je pouze informací pro učitele. Protože neprobíhá žádné hodnocení, neposkytne studentům zpětnou vazbu a možnost korekce. Tato varianta je zde tedy spíše pouze pro úplnost.
- **Prohlížení výsledků** – je-li tato volba „zatržena“, čtenáři nebudou schopni zobrazit výsledek tohoto testu. Může to být výhodné v situaci, kdy celá třída skládá současně stejný test. Rovněž pokud je test vyhodnocován učitelem, bude dobré *Prohlížení výsledků* zakázat až do té doby, než učitel test ohodnotí.
- **Blokování portálu** – po dobu práce na testu nebude možné přepínat se mezi moduly portálu, neboli student si nebude moci vyhledat informaci ve výukových materiálech kurzu.
- **Načasování uveřejnění a stažení** – tato možnost je podrobně rozebrána v kapitole 3.1.3 na straně 52. Je však na místě zdůraznit, že u testů má tato funkce velmi důležitou úlohu, protože umožňuje učiteli připravit si testy předem.

Název testu:	Test pro savce
Vyhodnocení testu:	<input checked="" type="checkbox"/> Po dokončení tento test vyhodnotit. (Funguje pouze při zapnutém automatickém hodnocení.)
Způsob hodnocení testu:	<input type="radio"/> Žádné vyhodnocení <input type="radio"/> Automatické vyhodnocení <input type="radio"/> Vyhodnocení učitelem <input checked="" type="radio"/> Vyhodnocení automatické i učitelem
Prohlížení výsledků:	<input type="checkbox"/> Čtenáři si nemohou prohlížet výsledky tohoto testu.
Blokování portálu:	<input checked="" type="checkbox"/> Po dobu práce na testu blokovat uživateli přechod do jiného modulu.
Datum uveřejnění:	2006-02-24
Čas uveřejnění:	8:00:00
Datum uzavření:	2006-02-24
Čas uzavření:	8:40:00
Nepřístupnost testu:	<input type="checkbox"/> Tento test je nepřístupný
Vyberte otázky pro test:	<input checked="" type="checkbox"/> Savci (Pravda / Nepravda) <input checked="" type="checkbox"/> Savci 2 (Výběr možností) <input checked="" type="checkbox"/> Savci 3 (Slovní odpověď) <input checked="" type="checkbox"/> Savci 4 (Vypočtená úloha)
Vytvořit test	

Obrázek 69: Vytvoření testu - parametry testu a výběr otázek

Test pro savce

3 . - Savci

1 . - Savci 2

4 . - Savci 3

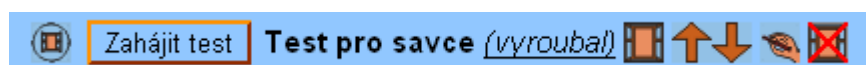
2 . - Savci 4

Vytvořit test

Obrázek 70: Vytvoření testu - pořadí otázek

Administrace testů

Pokud má uživatel dostatečná oprávnění, může provádět administraci testů pomocí ikon zobrazených u testu v seznamu testů:



Obrázek 71: Ikony pro administraci testu

Při editaci testu je možné měnit parametry testu, ale není možné změnit výběr otázek, které patří do testu. Takový zásah do struktury testu by narušil možnost zpětného prohlížení výsledků dosažených v testech.

Odstraněním testu se z databáze odstraní i všechny pokusy, které k testu přináležejí. Otázky, ze kterých je test složen, samozřejmě odstraněny nejsou.

3.5.2.1 Přístupová práva

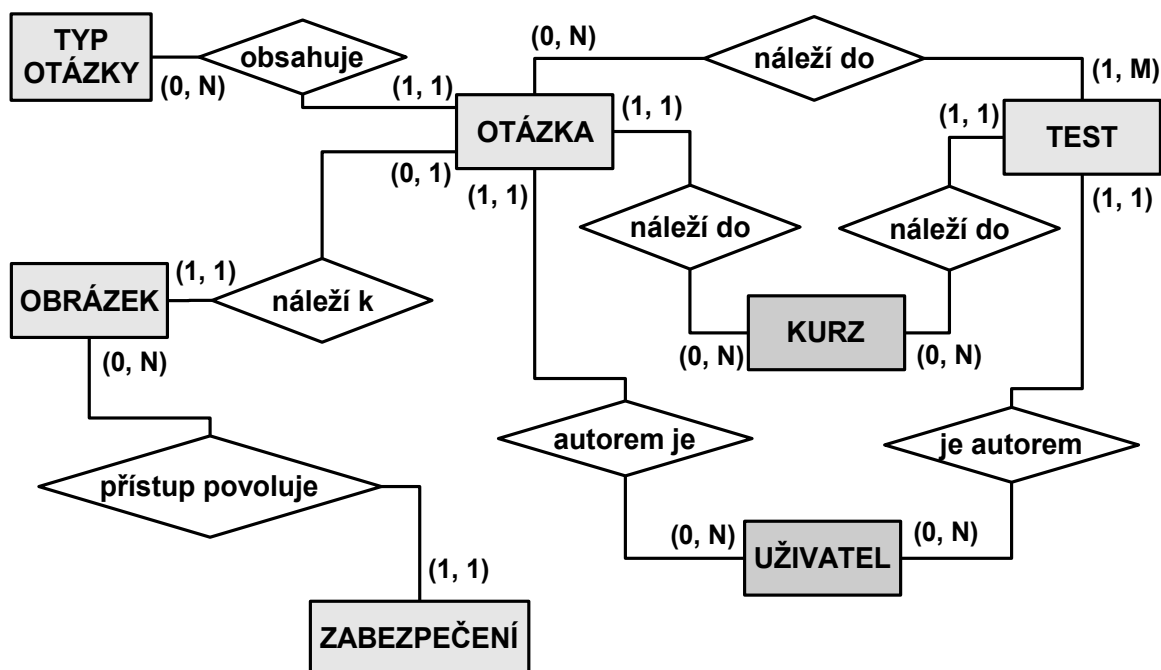
Při práci s testy jsou uživatelům přidělována oprávnění na základě role, kterou má v kurzu přidělenou. Přehled rolí a souvisejících oprávnění shrnuje tabulka 7.

	<i>Vlastník a redaktor</i>	<i>Příspěvateľ</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Vytvoření nového testu (s použitím libovolné otázky v kurzu)	X	X			
Změna pořadí testů	X				
Editace a výmaz jakéhokoli testu	X				
Editace a výmaz vlastního testu	X	X			

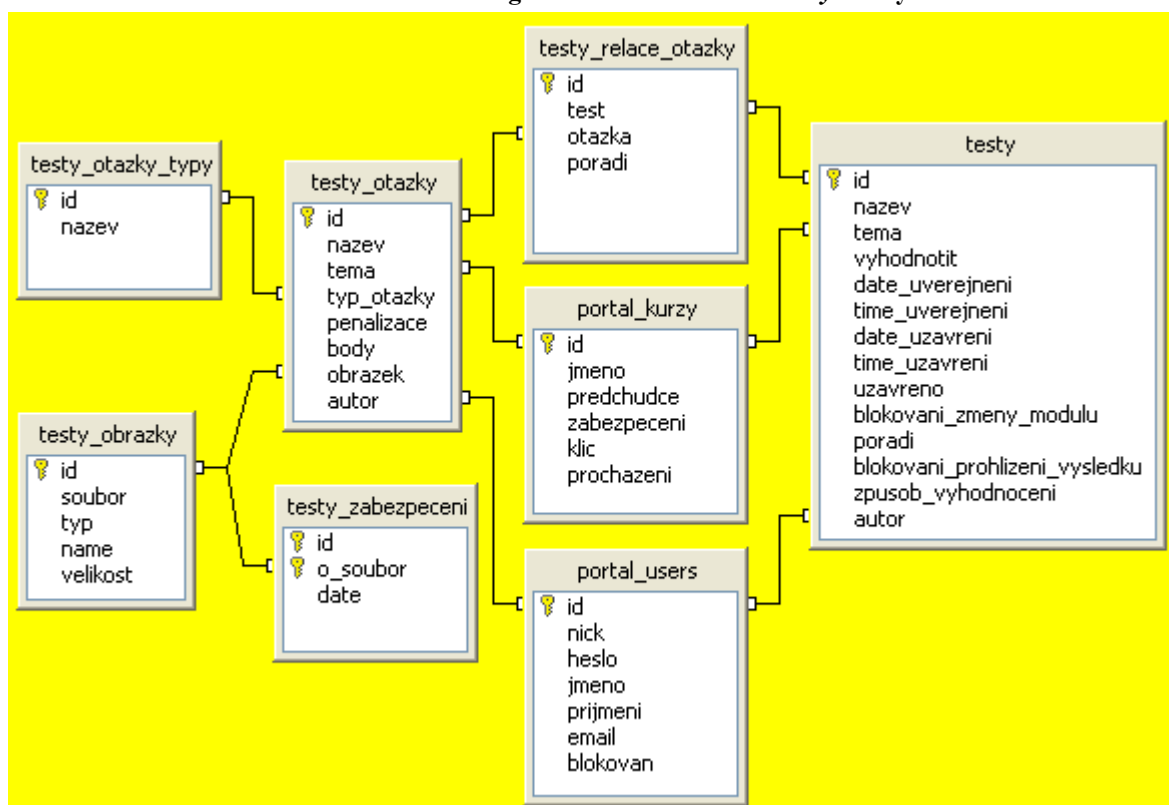
Tabulka 7: Oprávnění v modulu *Testy* – testy

3.5.2.2 Databáze

V této části uvedeme znázornění části databáze náležející modulu *Testy*, která uchovává informace o vytvořených testech. Hlavním entitním typem je zde *TEST*, kterému jsou (v relaci M:N) přiřazeny otázky (entity entitního typu *OTÁZKA*).



Obrázek 72: ER diagram databáze modulu *Testy* - testy



Obrázek 73: Znázornění tabulek, atributů a relací mezi tabulkami

3.5.3 Výsledky testů









Čtenář, který má příslušné oprávnění, může v kurzu absolvovat test. V souvislosti s tímto absolvováním se uchovávají informace o výsledku testu a o průběhu pokusu. Ve chvíli, kdy uživatel zahájí test, je tato akce zaznamenána do tabulky *testy_odpoved_pokus*. Student tedy nemůže zahájit test „nanečisto“ bez vědomí učitele. Tento záznam provádí metoda třídy *testy* (soubor *class_test.php*):

```
function zahajeni_testu($stab_odpoved_test, $id_test, $usr_id){
    //Ulozeni informaci o zahajeni testu do db
    $dotaz = "INSERT INTO `". $stab_odpoved_test
        . "` (`kdo`, `test`, `start_date`, `start_time`) VALUES ('"
        . $usr_id . "', '" . $id_test . "', '"
        . date("Y-m-d") . "', '" . date("H:i:s") . "')";
    session_register('test_pokus');
    $_SESSION['test_pokus'] = $this->sql_insert_return_id($dotaz);
}
```


Při korektním ukončení testu je k tomuto záznamu připojeno datum a čas dokončení jinou z metod stejné třídy. Učitel tak má informaci o celkovém času, který student strávil prací na testu.

```
function ukonceni_testu($stab_odpoved_test, $body){
    $dotaz = "UPDATE `". $stab_odpoved_test . "` SET "
        . "`finish_date`='" . date("Y-m-d") . "', "
        . "`finish_time`='" . date("H:i:s") . "', "
        . "`hodnoceni`='" . $body . "' "
        . "WHERE `id`='" . $_SESSION['test_pokus'] . "' LIMIT 1";
    $this->sql_dotaz($dotaz);
    session_unregister('test_pokus');
}
```

Pokud jsou k dispozici správné výsledky, je navíc vypočteno procento úspěšnosti studenta, které se ukládá do tabulky *testy_odpoved_pokus* a rovněž se zobrazuje v seznamu výsledků testů. V tomto seznamu je jméno uživatele, který test vypracoval, odkazem na podrobnosti o tomto uživateli.

Výsledky testu				
Test: Test pro savce				
Uživatel	Hodnocení	Zahájeno	Dokončeno	
  vyroubal (Michal Vyroubal)	0 %	2006-02-24, 09:52:31	Nedokončeno	
  vyroubal (Michal Vyroubal)	9 %	2006-02-23, 12:30:53	2006-02-23, 12:30:56	
  anonymní uživatel	9 %	2006-02-23, 12:29:22	2006-02-23, 12:29:25	
  testctenar (Tester Čtenář)	21 %	2006-02-17, 15:22:17	2006-02-17, 15:22:26	

Obrázek 74: Seznam výsledků testů

Ikonou  je možné zobrazit detaily výsledku – odpovědi na jednotlivé otázky. Při zobrazení těchto detailů může učitel doplnit ke každé otázce slovní hodnocení a vlastní

bodování či známkování. Příklad zobrazení odpovědi na jednu otázku je na obrázku 75 – v prvním sloupci je odpověď studenta, ve druhém výsledek automatického hodnocení testu a v posledním vyhodnocení učitelem (zobrazení těchto sloupců je závislé na tom, zda jsou uvedena vyhodnocení pro test povolena). Obrázek navíc ukazuje situaci, kdy učitel hodnotí jinak než automatické vyhodnocení, což může nastat například u otázky typu *Slovní odpověď*.

<p>Savci</p>  <p>Je na obrázku savec?</p> <p>Pravda</p>	<p>špatně Požadovaný výsledek: Nepravda Body: -0.5 (max: 3.5, penalizace: 0.5)</p>	<p>Správně</p> <p>Přiřazené body: 100</p> <p>Poznámka učitele: Savec je schovaný za autem :-)</p>
---	---	---

Obrázek 75: Zpracování odpovědi na jednu testovou otázku

Učitel může připojit závěrečné hodnocení celého testu:

<p><i>Celkem bodů:</i> 0.8 <i>Maximálně bodů:</i> 9 Hodnocení: 8.8%</p>	<p>Poznámka učitele: Tento test se Vám mimořádně povedl – jen tak dál.</p> <p>Závěrečné hodnocení: 1*</p> <p>Uložit hodnocení učitelem</p>
--	---

Obrázek 76: Závěrečné hodnocení testu

Pokud je to studentovi umožněno (záleží na nastavení testu), může si prohlížet výsledky svých testů včetně případných připojených komentářů učitele:

<p>Savci</p>  <p>Je na obrázku savec?</p> <p><input type="text" value="Pravda"/></p>	<p>špatně Požadovaný výsledek: Nepravda Body: -0.5 (max: 3.5, penalizace: 0.5)</p>	<p>správně Přiřazené body: 100 Poznámka učitele: Savec je schovaný za autem :-)</p>
--	---	--

Obrázek 77: Prohlížení výsledku testu studentem

3.5.3.1 Přístupová práva

Při práci s testy a výsledky testů mají uživatelé přidělena oprávnění na základě role, kterou mají v kurzu. Tabulka 8 poskytuje přehled o těchto oprávněních:

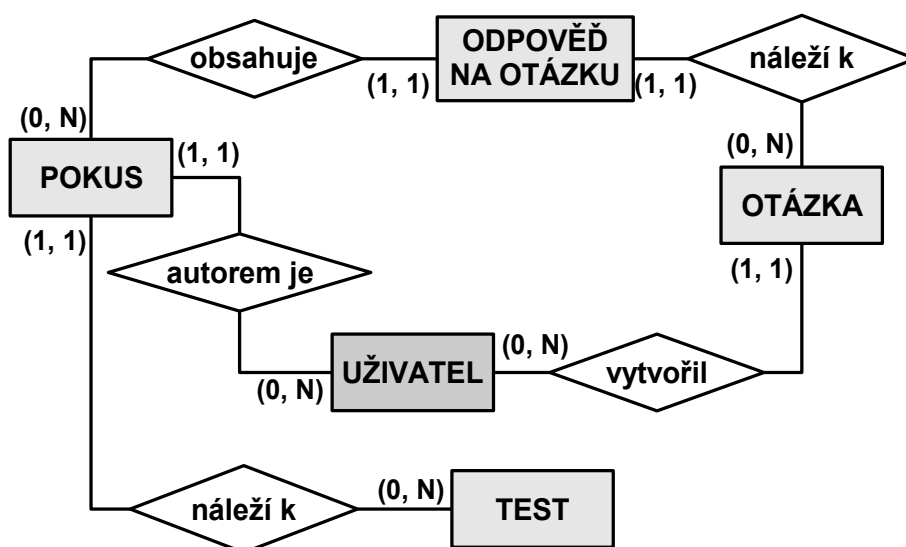
	<i>Vlastník a redaktor</i>	<i>Příspěvatel</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Absolvování (vyplnění) testu	X	X	X	X	
Výmaz výsledku u jakéhokoli testu	X				
Výmaz výsledku u vlastního testu	X	X			
Doplnění komentáře k výsledku u jakéhokoli testu	X				
Doplnění komentáře k výsledku u vlastního testu	X	X			
<i>Prohlížení výsledků absolvování testu:</i>					
Všech výsledků u jakéhokoli testu	X				
Všech výsledků u vlastního testu	X	X			
Vlastních výsledků	X	X	X		

Tabulka 8: Oprávnění v modulu *Testy* – výsledky testů

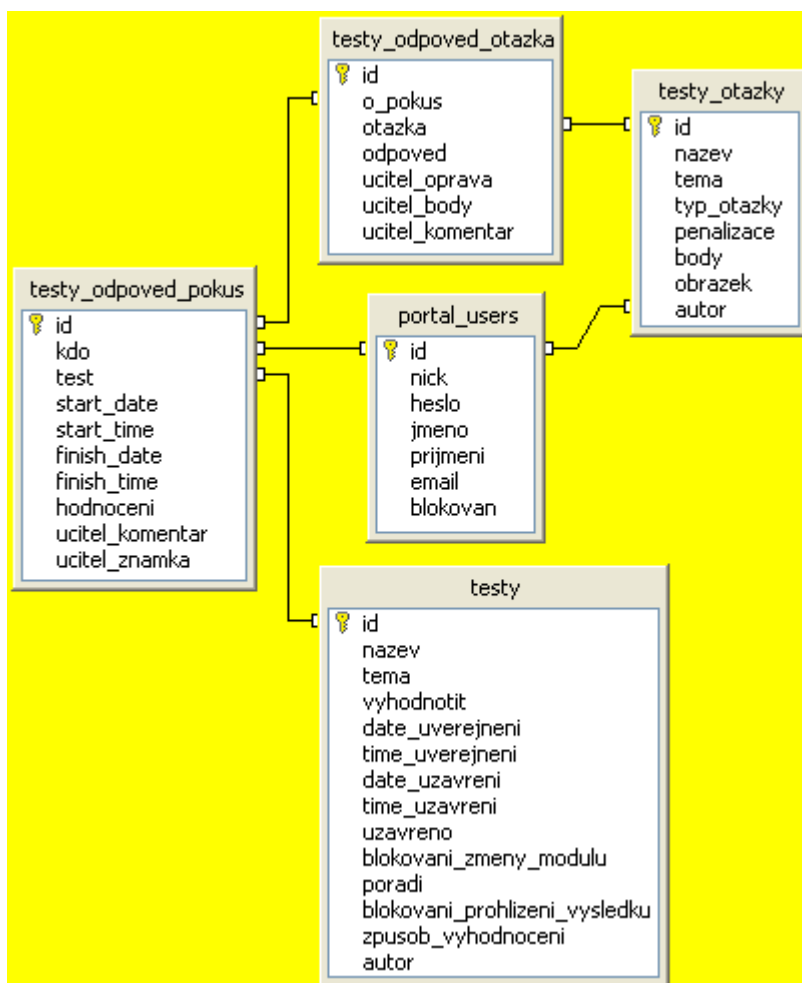
3.5.3.2 Databáze

V části databáze přináležející výsledkům testů, jsou důležité entitní typy *POKUS* (tabulka *testy_odpoved_pokus*) – představující samotný pokus o absolvování testu a s ním související informace – a entitní typ *ODPOVĚĎ NA OTÁZKU* (tabulka *testy_odpoved_otazka*) – který představuje konkrétní odpovědi studenta na jednotlivé otázky v testu. Do tabulky *testy_odpoved_otazka* se rovněž ukládá hodnocení připojené učitelem k otázkám a do tabulky *testy_odpoved_pokus* se ukládá hodnocení učitele k celému testu a výsledek automatického hodnocení pro test.

Jak je vidět z diagramu na obrázku 78, nemusí pokusu příslušet žádná odpověď na otázku. Tato situace nastane ve chvíli, kdy test není řádně ukončen a nejsou tudíž zaznamenány informace o dokončení testu (datum a čas dokončení) ani odpovědi na testové otázky.



Obrázek 78: ER diagram databáze modulu *Testy* - výsledky testů



Obrázek 79: Znáznornění tabulek, atributů a relací mezi tabulkami

3.6 Modul Lekce

Jak bylo ukázáno v předchozích kapitolách, je výukový obsah kurzu rozdělen do jednotlivých modulů. Student by mohl jen velmi těžko vytušit, jak spolu jednotlivé články, testy, soubory, atd. souvisejí a v jakém pořadí je má procházet. Tuto informaci mu poskytuje právě modul *Lekce*.

Modul *Lekce* umožňuje učitelům vytvořit lineární výukový program z objektů roztroušených v jednotlivých modulech – tento program může student procházet bez ohledu na to, ve kterém modulu je položka umístěna. Kurz může obsahovat libovolný počet lekcí a každá lekce může obsahovat libovolný počet položek odkazujících na objekty v jiných modulech. Učitel tedy může látku dobře strukturovat.

Vytvoření a administrace lekce

Při vytváření nové lekce je možné vytvořit komentář k lekci. Rovněž je možné načasovat uveřejnění a stažení lekce z portálu (tato možnost je popsána v kapitole 3.1.3 na straně 52).












Jméno lekce:	Základní aktivity uživatele
Popis lekce:	Aneb co potřebují vědět všichni (teda kromě dětí do 3 let věku - těm to neřekneme)
Datum uveřejnění:	2006-01-26
Čas uveřejnění:	08:00:00
Datum uzavření:	
Čas uzavření:	
Nepřístupnost lekce:	<input type="checkbox"/> Tato lekce je nepřístupná
Odeslat	

Obrázek 80: Vytvoření nové lekce


Administrace lekce se provádí obdobným způsobem jako například v modulu *Články*. Pokud je lekce odstraněna, jsou smazány i všechny související záznamy o přiřazených položkách – objekty, na které tyto položky ukazují do jednotlivých modulů, smazány nejsou.

Sestavení lekce a administrace položek

Pokud je lekce vytvořena, je možné na ni přidávat položky z jednotlivých modulů. Toto přidávání se provádí výběrem z vyklápěcího seznamu select, jak ukazuje obrázek 81.

  	Základní aktivity uživatele										
	Aneb co potřebují vědět všichni (teda kromě dětí do 3 let věku - těm to neřekneme)										
	    Úvodem omluva (Články)										
	    log-in-out a registrace (Články)										
	Přidat do této lekce:										
<table border="1"> <thead> <tr> <th>Články</th> <th>Testy</th> <th>Soubory</th> <th>Diskuse</th> </tr> </thead> <tbody> <tr> <td> Vyberte... Vyberte... Úvodem omluva log-in-out a registrace Správa všech uživatelů Správa uživatelů tématu </td> <td> Vyberte... Vložit </td> <td> Vyberte... Vložit </td> <td> Vyberte... Vložit </td> </tr> </tbody> </table>				Články	Testy	Soubory	Diskuse	Vyberte... Vyberte... Úvodem omluva log-in-out a registrace Správa všech uživatelů Správa uživatelů tématu	Vyberte... Vložit	Vyberte... Vložit	Vyberte... Vložit
Články	Testy	Soubory	Diskuse								
Vyberte... Vyberte... Úvodem omluva log-in-out a registrace Správa všech uživatelů Správa uživatelů tématu	Vyberte... Vložit	Vyberte... Vložit	Vyberte... Vložit								

Obrázek 81: Přidání položky do lekce

Pomocí administrativních ikon je možné měnit pořadí položek v lekci, nebo odstranit položku lekce. Editace položky (ikona ) v tomto případě představuje odkaz na editaci daného objektu (needituje se tedy položka lekce, ale přímo objekt (například článek), na který položka odkazuje).

3.6.1 Přístupová práva

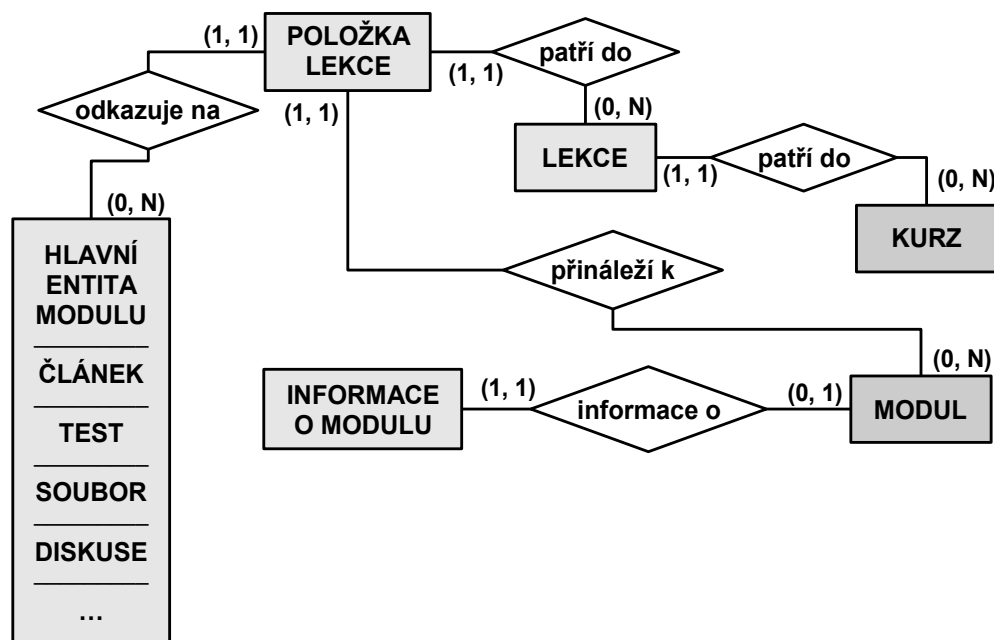
Nyní si shrňme, jaké aktivity jsou v modulu *Lekce* povolené uživatelům s jednotlivými rolemi:

	<i>Vlastník a redaktor</i>	<i>Příspěvatel</i>	<i>Čtenář přihlášený</i>	<i>Čtenář nepřihlášený</i>	<i>Bez přístupu</i>
Zobrazení lekce a položek v lekci	X	X	X	X	
Vytvoření nové lekce	X				
Změna pořadí lekcí	X				
Editace a výmaz lekce	X				
Přidání položky do lekce	X				
Změna pořadí položek v lekci	X				
Odstranění položky z lekce	X				

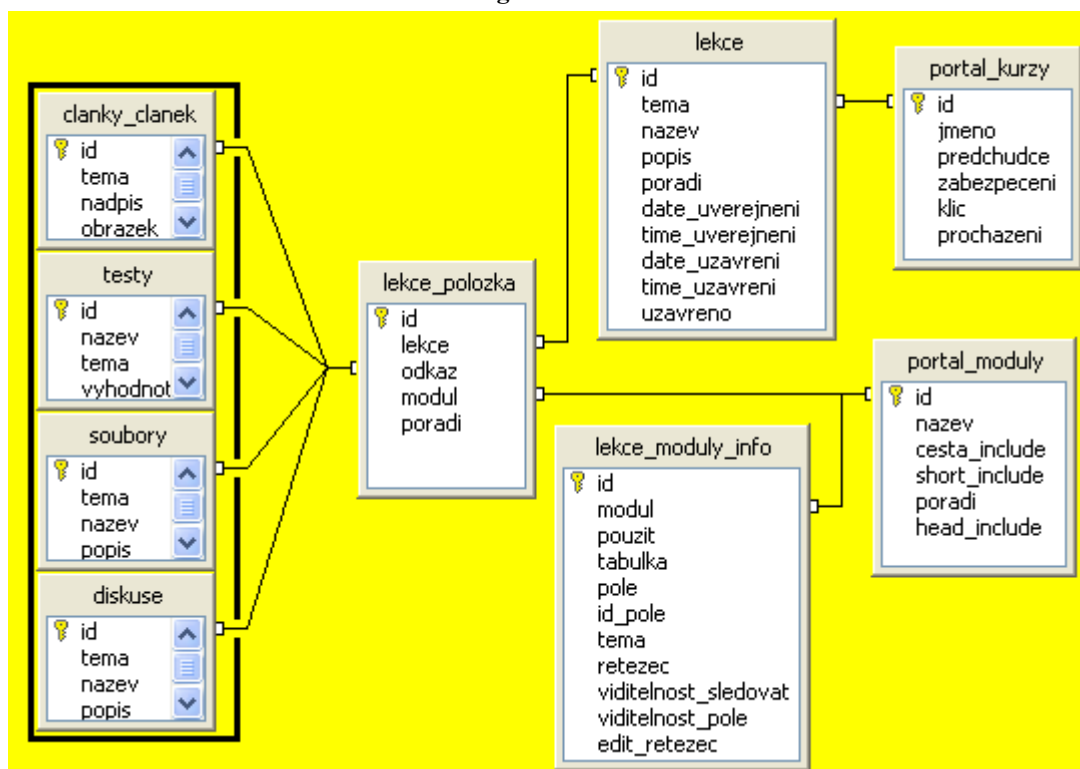
Tabulka 9: Oprávnění v modulu *Lekce*

3.6.2 Databáze

Na základě konvencí rozebraných v kapitole 1.2.3 nyní zakreslíme ER diagram databáze modulu *Soubory* a znázornění tabulek, atributů a relací.



Obrázek 82: ER diagram databáze modulu *Lekce*



Obrázek 83: Znázornění tabulek, atributů a relací mezi tabulkami

Ústředními entitními typy jsou zde *LEKCE* a *POLOŽKA LEKCE*. Entitní typ *LEKCE* je podobný jako například entitní typ *ČLÁNEK* z databáze modulu *Články*.

Zajímavé jsou pro nás vlastnosti entitního typu *POLOŽKA LEKCE* (tabulka *lekce_polozka*). Entitní typ *POLOŽKA LEKCE* je v relaci *přináleží k* s entitními typy *MODUL*. Tato relace popisuje, do kterého modulu daná položka odkazuje – zda jde o odkaz např. na článek nebo test. Relace *odkazuje na* spojuje entitní typ *POLOŽKA LEKCE* s hlavními entitními typy jednotlivých modulů – např. v modulu *Články* jde o entitní typ *ČLÁNEK* (tabulku *clanky_clanek*), která obsahuje základní informace o článku. Na základě relace *přináleží k* je tedy rozhodnuto, který z modulů se pro odkaz použije neboli ze které tabulky se budou brát hodnoty cizího klíče *odkaz*.

Entitní typ *INFORMACE O MODULU* (tabulka *lekce_moduly_info*) doplňuje k základním informacím o modulu, které poskytuje přímo portál, svoje vlastní informace. Uvedený entitní typ je v relaci s entitním typem *MODUL*. Kardinalita této relace je typu 1:1, ale členství entit *MODUL* je nepovinné – ne každý modul bude umísťovat svoje objekty jako položky lekce (například modul *Home*). O tomto entitním typu je více pojednáno v kapitole 3.6.4.

Data

Pro správné fungování jsou od začátku v databázi následující záznamy:

- tabulka *lekce_moduly_info* – údaje o modulech nutné pro jejich registraci do lekcí (bude vysvětleno dále)

3.6.3 Registrace modulu lekce v portálu

V modulech, které jsou začleněny v portálu, je zobrazen odkaz, který studenta vrátí zpět do modulu *Lekce*, aby tak student mohl dále systematicky pokračovat v procházení výukového obsahu. Aby takový odkaz mohl být v modulech vytvořen, je v souboru *include/config.php* definována konstanta obsahující název modulu *Lekce*:

```
| define("LEKCE_JMENO", 'Lekce');
```

Při inicializaci portálu, když jsou načítány moduly, je na základě této konstanty nalezen modul *Lekce* a jeho primární atribut je uložen do session proměnné (soubor *index.php* – proměnná *\$modul[1]* představuje jméno a *\$modul[0]* představuje id modulu, který je právě registrován):

```
| if($modul[1] == LEKCE_JMENO) {
|     session_register("modul_lekce_id");
|     $_SESSION["modul_lekce_id"] = $modul[0];
| }
```

Uvedená session proměnná je poté použita při sestavování odkazu zpět na lekci:

```
| echo "<p class=\"a_button\">"
|     "<a class=\"a_button\" href=\""
|         "$_SESSION[\"HOME_URL\"].\"index.php?modul=\""
|         "$_SESSION[\"modul_lekce_id\"]"
|         "&otec=\"$_SESSION[\"selected_tema\"]"
|     . "\">Zpět do kurzu</a></p>";
```

3.6.4 Registrace nového modulu do modulu *Lekce*

Pro možnost přidávat objekty z nového modulu, který je přidán do portálu, jakožto položky lekce, je nutné tento nový modul správně zaregistrovat nejprve v portálu a poté v samotném modulu *Lekce*.

Jakmile je nový modul zaregistrován do portálu (je přidán záznam do tabulky *portal_moduly* – více viz kapitolu 2.4.2 na straně 44), je k dispozici hodnota primárního klíče *id* tohoto modulu, kterou budeme používat jako cizí klíč v poli *modul* v tabulce *lekce_moduly_info* pro registraci modulu v *Lekcích*. Nyní si ukažme, jaké informace jsou uchovávány v tabulce *lekce_moduly_info* (viz obrázky 82 a 83 na str. 90):

<i>Atribut</i>	<i>Význam</i>	<i>Příklad hodnoty (pro modul Články)</i>
<i>id</i>	Primární atribut.	
<i>modul</i>	Cizí klíč z tabulky <i>portal_moduly</i> .	3
<i>pouzit</i>	Hodnota boolean určující, zda se modul bude používat pro přidávání položek do lekcí.	1
<i>tabulka</i>	Jméno hlavní tabulky modulu.	<i>mfp_clanky_clanek</i>
<i>pole</i>	Jméno atributu, který obsahuje název objektu v modulu (například <i>nadpis</i> článku).	<i>nadpis</i>
<i>id_pole</i>	Jméno klíčového atributu z hlavní tabulky modulu.	<i>id</i>
<i>tema</i>	Jméno atributu, který je v hlavní tabulce modulu cizím klíčem odkazujícím na entitu <i>KURZ</i> .	<i>tema</i>
<i>retezec</i>	Řetězec, který se připojí k URL, představující odkaz na zobrazení objektu. Hvězdička v tomto řetězci nahrazuje hodnotou primárního atributu příslušného objektu.	<i>clanky_show=1 &clanky_clanek=*</i>
<i>viditelnost_sledovat</i>	Boolean hodnota určující, zda má být u objektu sledována viditelnost, neboli uveřejnění. Pokud je viditelnost sledována, odkazy na neuveřejněné objekty se nebudou v položkách lekce zobrazovat.	1

<i>Atribut</i>	<i>Význam</i>	<i>Příklad hodnoty (pro modul Články)</i>
viditelnost_pole	Jména atributů obsahujících (v tomto pořadí) datum a čas uveřejnění, datum a čas stažení a ruční stažení objektu. Řetězec musí být v takovém tvaru, aby se dal rovnou používat v SQL dotazu jako názvy polí.	`date_uverejneni`, `time_uverejneni`, `date_uzavreni`, `time_uzavreni`, `uzavreno`
edit_retezec	Řetězec, který se připojí k URL, představující odkaz na zobrazení editace objektu. Jde o podobný princip jako u atributu retezec.	clanky_edit=1 &clanky_clanek=*

Kód PHP nahrazující znak hvězdička hodnotou primárního atributu

Následující kód nahradí při zobrazování položek lekce metodou `zobraz_lekce_tema` třídy `lekce` (soubor `moduly/lekce/class_lekce.php`) znak `"*"` hodnotou proměnné `$radek_polozky_lekce[2]`, která obsahuje primární klíč objektu, na který se odkazuje. Session proměnná, ve které je nahrazení provedeno, obsahuje hodnotu atributu `retezec` z tabulky `lekce_moduly_info`. Řetězec vzniklý uvedeným nahrazením se používá pro sestavení odkazu na objekt.

```
$polozka_retezec
= str_replace(
    "*",
    $radek_polozky_lekce[2],
    $_SESSION["modul"].$_SESSION["modul"].$poradi_modulu."id"
    ."lekce_retezec");
```

Podobně probíhá nahrazení při sestavování odkazu na editaci objektu.

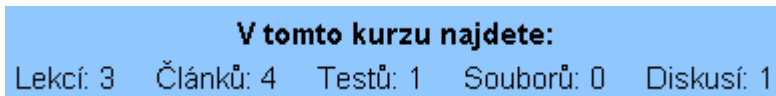
3.7 Modul Home

Modul *Home* se skládá z několika základních částí. První část je zjednodušením modulu *Články* – zjednodušení spočívá v tom, že se nezobrazuje seznam článků, ale přímo detaily článků, což je vhodné vzhledem k tomu, že v modulu *Home* se zobrazují základní informace o kurzu, které by měl návštěvník vidět přímo. Rovněž přístupová práva a struktura databáze odpovídají modulu *Články*. Touto funkcí se tedy nebudeme více zabývat, protože je popsána v kapitole 3.2 *Modul Články* na straně 59.

Informace o modulech v kurzu

Druhou částí modulu *Home* je výpis informací o kurzu napříč všemi moduly. Tento výpis se provádí pomocí include souboru, jehož jméno a cesta k němu jsou uvedeny jako hodnoty atributu `short_include` v tabulce `portal_moduly` (více viz kapitolu 2.4.2 *Tabulka pro registraci modulu v portálu* na str. 44). Je na autorovi modulu jakou informaci o obsahu modulu návštěvníkovi poskytne. Vzorové moduly, které jsou součástí této práce,

zahrnují do tohoto výstupu počet objektů, které jsou ve zvoleném kurzu přístupné pro aktuálně přihlášeného uživatele.



Obrázek 84: Zobrazení informací o kurzu

Jako příklad si uveďme kód souboru `moduly/clanky/short.php`, který poskytuje informaci o počtu přístupných článků v kurzu:

```
include 'config.php';
include 'class_clanky.php';
$new_clanek = new clanek;
//ochrana proti přímému zobrazení souboru
if($_SESSION["ma_prava_tema_final"] == "1"
    && $_SESSION["ma_prava_tema"] == "1"){
    echo "Článků: "
        . $new_clanek->pocet_clanku_tema($_SESSION['selected_tema']);
}
```

Tento soubor používá metodu `pocet_clanku_tema` třídy `clanek` (soubor `class_clanky.php`), která pomocí poměrně komplikované podmínky otestuje, zda je článek uveřejněn; návratovou hodnotou je počet článků, které jsou danému uživateli přístupné (jak je vidět z kódu, počet přístupných objektů se může lišit podle role, kterou má přihlášený uživatel v kurzu):

```
function pocet_clanku_tema($id_tema){
    $clanky = $this->nacti_clanky_tema($id_tema);
    $return = 0;
    if(sizeof($clanky) > 0){
        foreach($clanky as $radek){
            if(((($radek[6]." ".$radek[7] < date("Y-m-d H:i:s"))
                //Datum uverejneni je v minulosti
            )&&
                (($radek[8]." ".$radek[9] > date("Y-m-d H:i:s"))
                //Datum uzavreni je v budoucnosti
                || ($radek[8] == 0 && $radek[9] == 0)
                //Nebo neni zadano datum uzavreni
            )&&
                $radek[10] == "0"
                //Priznak UZAVRENO je false
            )
                //Clanek muze byt videt
                || ($_SESSION["tema_pravo"] != CTENAR
                    && isset($_SESSION["tema_pravo"])))
                //Uvedena omezeni se tykaji pouze ctenare
            ){
                ++$return;
            }
        }
    }
    return $return;
}
```

Informace o správcích kurzu

Modul *Home* zobrazuje, kteří uživatelé mají v kurzu roli *vlastník*, *redaktor* a *příspěvatel*. Informace o těchto uživatelích jsou pro studenty, kteří s portálem pracují, velmi důležité, aby měli možnost kontaktovat přímo osoby zodpovědné za kurz.

O tento kurz se starají:		
<i>Vlastníci</i>	<i>Redaktoři</i>	<i>Příspěvatelé</i>
<u>vyroubal</u> (Michal Vyroubal)	Žádný redaktor	Žádný příspěvatel

Obrázek 85: Zobrazení správců kurzu

Díky informacím této kapitoly můžeme efektivně využívat moduly začleněné do e-learningového portálu. Rovněž jsme mohli proniknout pod povrch a porozumět principům fungování modulů, což nám umožňuje moduly upravovat a zdokonalovat, případně vytvářet moduly nové.

4 Závěr

Díky výsledkům této diplomové práce se učitelům dostává do ruky nástroj pro tvorbu e-learningu. Učitel může učivo systematicky a logicky uspořádat (provést didaktickou transformaci učiva) do elektronických kurzů, které mu poslouží jako příprava na výuku. Díky elektronickým kurzům může učitel předložit studentům více různých obrazových prostředků, které výuku ožíví a přispějí k lepšímu pochopení a zapamatování předkládané látky. Dalším přínosem je možnost efektivně opakovat, procvičovat a testovat s použitím e-learningového portálu, což umožní studentům lepší sebekontrolu učení po probrání určitého tématického celku učiva. Učitel může rovněž lépe organizovat a řídit domácí přípravu studentů, protože elektronický kurz je přístupný stále.

Samozřejmě výuka spočívající pouze v procházení elektronických kurzů by potlačovala důležité podněty vyplývající z interakce učitel – student a deformoval by se rozvoj důležitých sociálních a komunikativních dovedností studenta. Je tedy zřejmé, že použití e-learningu nemůže nahradit klasickou výuku učitelem, ale má ji oživit, udělat zajímavější a přidat jí další rozměr.

Diplomová práce postupně předložila informace o e-learningu a návrh a implementaci e-learningového portálu. Práce dále obsahuje podrobný rozbor databáze, kterou e-learningový portál používá, a popis pracovního prostředí portálu. Nástroje, které byly pro implementaci portálu použity, jsou popsány v úvodní části práce, kde je rovněž provedena úvaha o jejich vhodnosti a případných alternativách. Diplomová práce dále obsahuje implementaci portálu pomocí PHP – kompletní zdrojový kód portálu. Cíle diplomové práce tedy bylo dosaženo.

Pro možnost vyzkoušení funkčnosti portálu je jeho testovací verze zprovozněna na adrese:

<http://vyrb1.hyperlinx.cz/diplomka/http/>

Portál, který byl v této práci vytvořen, musí být samozřejmě dále rozvíjen a rozšiřován. Vytvoření dalších modulů – například modul pro zadávání úkolů studentům a pro jejich následné odevzdání a kontrolu učitelem – je možné, protože diplomová práce velmi podrobně popisuje jak na to.

5 Literatura

- [1] Ministerstvo školství, mládeže a tělovýchovy. *Státní informační politika ve vzdělávání* [online]. Poslední aktualizace 21. dubna 2006 [cit. 2006-05-01]. <http://www.msmt.cz/_DOMEK/?CAI=2412>.
- [2] Kontis s. r. o. *E-learning* [online]. [cit. 2006-02-12]. <<http://www.e-learn.cz/>>.
- [3] POKORNÝ, J. *Konstrukce databázových systémů*. Praha: ČVUT, 2001.
- [4] POKORNÝ, J. *Databázové systémy a jejich použití v informačních systémech*. Praha: Academia, 1992.
- [5] STRAKA, M. *Vývoj databázových aplikací*. Praha: Grada, 1992.
- [6] SOUKUP, P. *Aplikace databázových technologií v digitální kartografii* [online]. Praha: září 2001. [cit. 2006-01-17]. <<http://gama.fsv.cvut.cz/~soukup/dis/>>.
- [7] *Konceptuální datový model* [online]. Poslední aktualizace 7. září 2004 [cit. 2006-01-17]. <<http://www.owebu.cz/database/vypis.php?clanek=298>>.
- [8] W3C. *Markup Validation Service* [online]. Poslední aktualizace 20. února 2006 [cit. 2006-03-15]. <<http://validator.w3.org/>>.
- [9] W3C. *Cascading Style Sheets, level 2 – CSS2 Specification* [online]. Uvolněno 12. května 1998 [cit. 2006-05-01]. <<http://www.w3.org/TR/REC-CSS2/>>.
- [10] SATRAPA, P. *World-Wide Web pro čtenáře, autory a misionáře*. Neokortex, 1996.
- [11] PEXA, P. *Tvorba WWW a WAP*. České Budějovice: Kopp, 2001.
- [12] The PHP Group. *Manuál PHP* [online]. Poslední aktualizace 20. února 2005 [cit. 2006-04-18]. <<http://www.php.net/manual/cs/>>.
- [13] KOSEK, J. *PHP – tvorba interaktivních internetových aplikací*. Praha: GRADA Publishing, 1999.
- [14] SKLAR, D. – TRACHTENBERG, A. *PHP Cookbook*. O'Reilly, 2002.
- [15] LERDORF, R. – TATROE, K. *Programming PHP*. O'Reilly, 2002.
- [16] WILLIAMS, H. – LANE, D. *PHP a MySQL – vytváříme webové databázové aplikace*. Praha: ComputerPress, 2002.

Příloha 1 – Zdrojový kód na CD

Na přiloženém CD je umístěn kompletní zdrojový kód portálu a popsanych modulů.

Příloha 2 – MySQL databáze

Portál

Struktura

```
CREATE TABLE "mfp_portal_moduly" (  
  "id" int(11) NOT NULL,  
  "nazev" text NOT NULL,  
  "cesta_include" text NOT NULL,  
  "short_include" text NOT NULL,  
  "poradi" tinyint(4) NOT NULL default '0',  
  "head_include" tinytext NOT NULL,  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_portal_kurzy" (  
  "id" int(11) NOT NULL,  
  "jmeno" tinytext NOT NULL,  
  "predchudce" int(11) NOT NULL default '0',  
  "zabezpeceni" tinyint(4) NOT NULL default '0',  
  "klic" tinytext NOT NULL,  
  "prochazeni" tinyint(1) NOT NULL default '1',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_portal_kurzy_oblibene" (  
  "id" int(11) NOT NULL,  
  "kdo" int(11) NOT NULL default '0',  
  "co" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_portal_kurzy_prava" (  
  "id" tinyint(4) NOT NULL,  
  "kam" tinyint(4) NOT NULL default '0',  
  "kdo" tinyint(4) NOT NULL default '0',  
  "pravo" tinyint(4) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_portal_users" (  
  "id" int(11) NOT NULL,  
  "nick" text NOT NULL,  
  "heslo" text NOT NULL,  
  "jmeno" text NOT NULL,  
  "prijmeni" text NOT NULL,  
  "email" tinytext NOT NULL,  
  "blokovan" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);
```

Data

```
INSERT INTO `mfp_portal_moduly`
VALUES (1, 'Home', 'moduly/home/hlavni.php',
'', 0, 'moduly/home/head.php');
INSERT INTO `mfp_portal_moduly`
VALUES (2, 'Lekce', 'moduly/lekce/hlavni.php',
'moduly/lekce/short.php', 1, 'moduly/lekce/head.php');
INSERT INTO `mfp_portal_moduly`
VALUES (3, 'Články', 'moduly/clanky/hlavni.php',
'moduly/clanky/short.php', 2, 'moduly/clanky/head.php');
INSERT INTO `mfp_portal_moduly`
VALUES (4, 'Testy', 'moduly/testy/hlavni.php',
'moduly/testy/short.php', 3, 'moduly/testy/head.php');
INSERT INTO `mfp_portal_moduly`
VALUES (5, 'Soubory', 'moduly/soubory/hlavni.php',
'moduly/soubory/short.php', 4, 'moduly/soubory/head.php');
INSERT INTO `mfp_portal_moduly`
VALUES (6, 'Diskuse', 'moduly/diskuse/hlavni.php',
'moduly/diskuse/short.php', 5, 'moduly/diskuse/head.php');

INSERT INTO `mfp_portal_kurzy`
VALUES (1, 'Kořenový kurz', 0, 0, '', 1);

INSERT INTO `mfp_portal_kurzy_prava`
VALUES (1, 1, 1, 10);

INSERT INTO `mfp_portal_users`
VALUES (1, 'root', '955db0b81ef1989b4a4dfeae8061a9a6',
'Hlavní', 'Uživatel', 'root@portal.cz', 0);
```

Modul Články

```
CREATE TABLE "mfp_clanky_clanek" (
  "id" int(11) NOT NULL,
  "tema" int(11) NOT NULL default '0',
  "nadpis" text NOT NULL,
  "obrazek" int(11) NOT NULL default '0',
  "shrnuti" text NOT NULL,
  "clanek" text NOT NULL,
  "date_uverejneni" date NOT NULL default '0000-00-00',
  "time_uverejneni" time NOT NULL default '00:00:00',
  "date_uzavreni" date NOT NULL default '0000-00-00',
  "time_uzavreni" time NOT NULL default '00:00:00',
  "uzavreno" tinyint(1) NOT NULL default '0',
  "poradi" tinyint(4) NOT NULL default '0',
  "autor" int(11) NOT NULL default '1',
  PRIMARY KEY ("id")
);
```

```
CREATE TABLE "mfp_clanky_obrazky" (  
  "id" int(11) NOT NULL,  
  "soubor" longblob NOT NULL,  
  "typ" text NOT NULL,  
  "name" text NOT NULL,  
  "velikost" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_clanky_zabezpeceni" (  
  "id" tinytext NOT NULL,  
  "o_soubor" int(11) NOT NULL default '0',  
  "date" datetime NOT NULL default '0000-00-00 00:00:00'  
);
```

Modul Soubory

```
CREATE TABLE "mfp_soubory" (  
  "id" int(11) NOT NULL,  
  "tema" int(11) NOT NULL default '0',  
  "nazev" tinytext NOT NULL,  
  "popis" text NOT NULL,  
  "o_soubor" int(11) NOT NULL default '0',  
  "poradi" int(11) NOT NULL default '0',  
  "date_uverejneni" date NOT NULL default '0000-00-00',  
  "time_uverejneni" time NOT NULL default '00:00:00',  
  "date_uzavreni" date NOT NULL default '0000-00-00',  
  "time_uzavreni" time NOT NULL default '00:00:00',  
  "uzavreno" tinyint(1) NOT NULL default '0',  
  "autor" int(11) NOT NULL default '1',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_soubory_soubory" (  
  "id" int(11) NOT NULL,  
  "soubor" longblob NOT NULL,  
  "typ" text NOT NULL,  
  "name" text NOT NULL,  
  "velikost" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_soubory_zabezpeceni" (  
  "id" tinytext NOT NULL,  
  "o_soubor" int(11) NOT NULL default '0',  
  "date" datetime NOT NULL default '0000-00-00 00:00:00'  
);
```

Modul Diskuse

```
CREATE TABLE "mfp_diskuse" (  
  "id" int(11) NOT NULL,  
  "tema" int(11) NOT NULL default '0',  
  "nazev" tinytext NOT NULL,  
  "popis" text NOT NULL,  
  "poradi" int(11) NOT NULL default '0',  
  "date_uverejneni" date NOT NULL default '0000-00-00',  
  "time_uverejneni" time NOT NULL default '00:00:00',  
  "date_uzavreni" date NOT NULL default '0000-00-00',  
  "time_uzavreni" time NOT NULL default '00:00:00',  
  "uzavreno" tinyint(1) NOT NULL default '0',  
  "autor" int(11) NOT NULL default '0',  
  "read_only" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "mfp_diskuse_prispevek" (  
  "id" int(11) NOT NULL,  
  "diskuse" int(11) NOT NULL default '0',  
  "otec" int(11) NOT NULL default '0',  
  "autor" int(11) NOT NULL default '0',  
  "nadpis" tinytext NOT NULL,  
  "text" text NOT NULL,  
  PRIMARY KEY ("id")  
);
```

Modul Testy

Struktura

```
CREATE TABLE "mfp_testy" (  
  "id" int(11) NOT NULL,  
  "nazev" text NOT NULL,  
  "tema" int(11) NOT NULL default '0',  
  "vyhodnotit" tinyint(1) NOT NULL default '1',  
  "date_uverejneni" date NOT NULL default '0000-00-00',  
  "time_uverejneni" time NOT NULL default '00:00:00',  
  "date_uzavreni" date NOT NULL default '0000-00-00',  
  "time_uzavreni" time NOT NULL default '00:00:00',  
  "uzavreno" tinyint(1) NOT NULL default '0',  
  "blokovani_zmeny_modulu" tinyint(1) NOT NULL default '1',  
  "poradi" tinyint(4) NOT NULL default '0',  
  "blokovani_prohlizeni_vysledku" tinyint(1) NOT NULL default '0',  
  "způsob_vyhodnoceni" tinytext NOT NULL,  
  "autor" int(11) NOT NULL default '1',  
  PRIMARY KEY ("id")  
);
```



```
CREATE TABLE "mfp_testy_obrazky" (  
  "id" int(11) NOT NULL,  
  "soubor" longblob NOT NULL,  
  "typ" text NOT NULL,  
  "name" text NOT NULL,  
  "velikost" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_odpoved_otazka" (  
  "id" int(11) NOT NULL,  
  "o_pokus" int(11) NOT NULL default '0',  
  "otazka" int(11) NOT NULL default '0',  
  "odpoved" text NOT NULL,  
  "ucitel_oprava" tinyint(1) NOT NULL default '0',  
  "ucitel_body" double NOT NULL default '0',  
  "ucitel_komentar" text NOT NULL,  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_odpoved_pokus" (  
  "id" int(11) NOT NULL,  
  "kdo" int(11) NOT NULL default '0',  
  "test" int(11) NOT NULL default '0',  
  "start_date" date NOT NULL default '0000-00-00',  
  "start_time" time NOT NULL default '00:00:00',  
  "finish_date" date NOT NULL default '0000-00-00',  
  "finish_time" time NOT NULL default '00:00:00',  
  "hodnoceni" float NOT NULL default '0',  
  "ucitel_komentar" text NOT NULL,  
  "ucitel_znamka" tinytext NOT NULL,  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky" (  
  "id" int(11) NOT NULL,  
  "nazev" text NOT NULL,  
  "tema" int(11) NOT NULL default '0',  
  "typ_otazky" int(11) NOT NULL default '0',  
  "penalizace" double NOT NULL default '0',  
  "body" double NOT NULL default '0',  
  "obrazek" int(11) NOT NULL default '0',  
  "autor" int(11) NOT NULL default '1',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky_typy" (  
  "id" int(11) NOT NULL,  
  "nazev" text NOT NULL,  
  PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "mfp_testy_otazky_typy_moznosti_vyber" (  
  "id" int(11) NOT NULL,  
  "o_zadani" int(11) NOT NULL default '0',  
  "text" text NOT NULL,  
  "spravne" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky_typy_moznosti_zadani" (  
  "id" int(11) NOT NULL,  
  "o_otazka" int(11) NOT NULL default '0',  
  "otazka" text NOT NULL,  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky_typy_pn" (  
  "id" int(11) NOT NULL,  
  "o_otazka" int(11) default NULL,  
  "otazka" text NOT NULL,  
  "spravne" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky_typy_slovni_odpoved" (  
  "id" int(11) NOT NULL,  
  "o_otazka" int(11) NOT NULL default '0',  
  "otazka" text NOT NULL,  
  "spravne" text NOT NULL,  
  "case_sensitive" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_otazky_typy_vypocet" (  
  "id" int(11) NOT NULL,  
  "o_otazka" int(11) NOT NULL default '0',  
  "otazka" text NOT NULL,  
  "spravne" double NOT NULL default '0',  
  "tolerance" double NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_relace_otazky" (  
  "id" int(11) NOT NULL,  
  "test" int(11) NOT NULL default '0',  
  "otazka" int(11) NOT NULL default '0',  
  "poradi" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_testy_zabezpeceni" (  
  "id" tinytext NOT NULL,  
  "o_soubor" int(11) NOT NULL default '0',  
  "date" datetime NOT NULL default '0000-00-00 00:00:00'  
);
```

Data

```
INSERT INTO `mfp_testy_otazky_typy`  
VALUES (1, 'Pravda / Nepravda');  
INSERT INTO `mfp_testy_otazky_typy`  
VALUES (2, 'Vypočtená úloha');  
INSERT INTO `mfp_testy_otazky_typy`  
VALUES (3, 'Výběr možností');  
INSERT INTO `mfp_testy_otazky_typy`  
VALUES (4, 'Slovní odpověď');
```

Modul Lekce

Struktura

```
CREATE TABLE "mfp_lekce" (  
  "id" int(11) NOT NULL,  
  "tema" int(11) NOT NULL default '0',  
  "nazev" tinytext NOT NULL,  
  "popis" text NOT NULL,  
  "poradi" int(11) NOT NULL default '0',  
  "date_uverejneni" date NOT NULL default '0000-00-00',  
  "time_uverejneni" time NOT NULL default '00:00:00',  
  "date_uzavreni" date NOT NULL default '0000-00-00',  
  "time_uzavreni" time NOT NULL default '00:00:00',  
  "uzavreno" tinyint(1) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_lekce_moduly_info" (  
  "id" int(11) NOT NULL,  
  "modul" int(11) NOT NULL default '0',  
  "pouzit" tinyint(1) NOT NULL default '0',  
  "tabulka" tinytext NOT NULL,  
  "pole" tinytext NOT NULL,  
  "id_pole" tinytext NOT NULL,  
  "tema" tinytext NOT NULL,  
  "retezec" tinytext NOT NULL,  
  "viditelnost_sledovat" tinyint(1) NOT NULL default '0',  
  "viditelnost_pole" tinytext NOT NULL,  
  "edit_retezec" tinytext NOT NULL,  
  PRIMARY KEY ("id")  
);  
  
CREATE TABLE "mfp_lekce_polozka" (  
  "id" int(11) NOT NULL,  
  "lekce" int(11) NOT NULL default '0',  
  "odkaz" int(11) NOT NULL default '0',  
  "modul" int(11) NOT NULL default '0',  
  "poradi" int(11) NOT NULL default '0',  
  PRIMARY KEY ("id")  
);
```

Data

```

INSERT INTO `mfp_lekce_moduly_info`
VALUES (1, 3, 1, 'mfp_clanky_clanek', 'nadpis', 'id', 'tema',
       'clanky_show=1&clanky_clanek=*', 1, '`date_uverejneni`,
       `time_uverejneni`, `date_uzavreni`, `time_uzavreni`, `uzavreno`,
       'clanky_edit=1&clanky_clanek=*');
INSERT INTO `mfp_lekce_moduly_info`
VALUES (2, 5, 1, 'mfp_soubory', 'nazev', 'id', 'tema',
       'soubor_show=1&soubory_id=*', 1, '`date_uverejneni`,
       `time_uverejneni`, `date_uzavreni`, `time_uzavreni`, `uzavreno`,
       'soubor_edit=1&soubory_id=*');
INSERT INTO `mfp_lekce_moduly_info`
VALUES (3, 4, 1, 'mfp_testy', 'nazev', 'id', 'tema',
       'show_test=1&test=*', 1, '`date_uverejneni`,
       `time_uverejneni`, `date_uzavreni`, `time_uzavreni`, `uzavreno`,
       'test_edit_test=1&test=*');
INSERT INTO `mfp_lekce_moduly_info`
VALUES (4, 6, 1, 'mfp_diskuse', 'nazev', 'id', 'tema',
       'diskuse_show=1&diskuse_id=*', 1, '`date_uverejneni`,
       `time_uverejneni`, `date_uzavreni`, `time_uzavreni`, `uzavreno`,
       'diskuse_edit=1&diskuse_id=*');

```

Modul Home

```

CREATE TABLE "mfp_home_clanek" (
  "id" int(11) NOT NULL,
  "tema" int(11) NOT NULL default '0',
  "nadpis" text NOT NULL,
  "obrazek" int(11) NOT NULL default '0',
  "shrnuti" text NOT NULL,
  "clanek" text NOT NULL,
  "date_uverejneni" date NOT NULL default '0000-00-00',
  "time_uverejneni" time NOT NULL default '00:00:00',
  "date_uzavreni" date NOT NULL default '0000-00-00',
  "time_uzavreni" time NOT NULL default '00:00:00',
  "uzavreno" tinyint(1) NOT NULL default '0',
  "poradi" tinyint(4) NOT NULL default '0',
  "autor" int(11) NOT NULL default '1',
  PRIMARY KEY ("id")
);

CREATE TABLE "mfp_home_clanky_obrazky" (
  "id" int(11) NOT NULL,
  "soubor" longblob NOT NULL,
  "typ" text NOT NULL,
  "name" text NOT NULL,
  "velikost" int(11) NOT NULL default '0',
  PRIMARY KEY ("id")
);

CREATE TABLE "mfp_home_zabezpeceni" (
  "id" tinytext NOT NULL,
  "o_soubor" int(11) NOT NULL default '0',
  "date" datetime NOT NULL default '0000-00-00 00:00:00'
);

```

Příloha 3 – Screenshoty portálu

Všechny screenshoty portálu byly pořízeny v rozlišení 1024 x 768.

Úvodní strana portálu

Mgr. MICHAL VYRUBAL

Vítejte!

Přihlašte se svým uživatelským jménem a heslem.

[Přihlásit se](#)

Pokud nemáte uživatelské jméno, zaregistrujte se jako uživatelé portálu.

[Zaregistrovat se](#)

Trváte-li na své anonymitě, zkuste štěstí jako anonymní nepřihlášený uživatel.

Svůj kurz si vyberte vlevo v seznamu kurzů.

Adresa tohoto příspěvku: <http://server6/diplomka/index.php?modul=1&otec=1>

Copyright © 2005 Mgr. Michal Vyrubal. Všechna práva vyhrazena.

W3C XHTML 1.0 W3C CSS

Skrýt seznam kurzů

Role ve vybraném kurzu:
Čtenář (nepřihlášen)

Seznam kurzů

Vyrubalův e-learning

Čas jako návod

1. portál
2. moduly
- ...

Zátěžová herna
matematika

...

Zobrazit všechny kurzy

Přihlásit se - Zaregistrovat se
pátek, 24. únor 2006, 10:29:24

Soubor Úpravy Zobrazit Přejít Zálžky Nástroje Nápořádá

Zobrazení lekcí v kurzu

The screenshot shows a web browser window with the address bar displaying 'Vyrubalův e-learning - Mozilla Firefox'. The page has a blue header with the site name 'Mgr. MICHAL VYRUBAL' in large orange letters. Below the header is a navigation menu with links: Home, Lekce, Články, Testy, Soubory, and Diskuse. A sidebar on the left contains links: Soubor, Úpravy, Zobrazit, Přijít, Zálžky, Nástroje, and Nápořádá. The main content area is divided into two columns. The left column contains a list of lessons: 'Skrýt seznam kurzů', 'Vyrubalův elearning > Cosi jako návod > 1. portál > 2. uživatelé', 'Přejít k nadřazenému kurzu', 'Seznam kurzů', and '2. uživatelé'. The right column contains a list of lessons: 'Základní aktivity uživatele', 'Aktivity správce portálu', and 'Poslední lekce'. The 'Základní aktivity uživatele' lesson is selected and expanded, showing a description: 'Ánež co potřebují vědět všichni (teda kromě dětí do 3 let věku - těm to neřekneme)' and a list of links: 'Úvodem omluva (Články)', 'log-in-out a registrace (Články)', 'Správa všech uživatelů (Články)', and 'Správa uživatelů tématu (Články)'. The 'Aktivity správce portálu' lesson is also expanded, showing a description: 'Toto již není pro všechny uživatele, ale pouze pro ty ... kdo se dříve se správou portálu' and a list of links: 'Správa všech uživatelů (Články)', 'Správa uživatelů tématu (Články)', 'Uživatelé portálu (Testy)', and 'Tak co vy na to (Diskuse)'. The 'Poslední lekce' lesson is also expanded, showing a description: 'Opět si můžete vyzkoušet, co jste se dozvěděli a zadiskutovat si.' and a list of links: 'Uživatelé portálu (Testy)' and 'Tak co vy na to (Diskuse)'. At the bottom of the page, there is a footer with the text: 'Adresa tohoto příspěvku: http://server6/diplomka/index.php?modul=2&otec=7', 'Copyright © 2005 Mgr. Michal Vyrubal. Všechna práva vyhrazena.', and three logos: 'W3C XHTML 1.0', 'W3C CSS', and 'W3C'.

Mgr. MICHAL VYRUBAL

Home Lekce Články Testy Soubory Diskuse

Přihlásit se - Zaregistrovat se
pátek, 24. únor 2006, 10:31:00

Soubor Úpravy Zobrazit Přijít Zálžky Nástroje Nápořádá

Skrýt seznam kurzů
Vyrubalův elearning > Cosi jako návod > 1. portál > 2. uživatelé

Role ve vybraném kurzu:
Čtenář (nepřihlášen)

Přejít k nadřazenému kurzu

Seznam kurzů

2. uživatelé

Zobrazit všechny kurzy

Seřadit lekce: vzestupně sestupně

Základní aktivity uživatele
Ánež co potřebují vědět všichni (teda kromě dětí do 3 let věku - těm to neřekneme)
[Úvodem omluva \(Články\)](#)
[log-in-out a registrace \(Články\)](#)

Aktivity správce portálu
Toto již není pro všechny uživatele, ale pouze pro ty ... kdo se dříve se správou portálu
[Správa všech uživatelů \(Články\)](#)
[Správa uživatelů tématu \(Články\)](#)

Poslední lekce
Opět si můžete vyzkoušet, co jste se dozvěděli a zadiskutovat si.
[Uživatelé portálu \(Testy\)](#)
[Tak co vy na to \(Diskuse\)](#)

Adresa tohoto příspěvku: <http://server6/diplomka/index.php?modul=2&otec=7>
Copyright © 2005 Mgr. Michal Vyrubal. Všechna práva vyhrazena.

W3C XHTML 1.0 W3C CSS W3C

Administrace lekcí v kurzu

The screenshot shows the 'Vyroubalův e-learning' portal in a Mozilla Firefox browser. The interface is divided into several sections for course administration.

Top Navigation: Home, Lekce, Články, Testy, Soubory, Diskuse. A status bar at the top right indicates 'Jste vyroubal/- Odhláší se - Změnit údaje' and the date 'pátek, 24. únor 2006, 10:36:47'.

Left Sidebar:

- Skrýt seznam kurzů**: Vyroubalův elearning > Cosi jako návod > 1. portál > 2. uživatelé
- Role ve vybraném kurzu:** Zohledněné vlastnictví
- Přejít k nadřazenému kurzu**: Editace uživatelů tohoto kurzu, Přidat sem kurz, Upravit tento kurz, Smazat tento kurz
- Seznam kurzů 2. uživatelé**: Zobrazit všechny kurzy, Oblíbené kurzy, Přidat tento kurz k oblíbeným, Kurzy, ke kterým mám právo

Main Content Area:

- Nová lekce**: Seřadit lekce: vzestupně sestupně
- Základní aktivity uživatele**: Aneb co potřebují vědět všichni (teda kromě dětí do 3 let věku - těm to neřekneme). Includes links for Úvodem omluva (Články) and log-in-out a registrace (Články). Below is a table for adding activities:

Články	Testy	Soubory	Diskuse
Vyberte...	Vyberte...	Vyberte...	Vyberte...
Vložit	Vložit	Vložit	Vložit
- Aktivity správců portálu**: Toto již není pro všechny uživatele, ale pouze pro ty ... kdo se dříve se správou portálu. Includes links for Správa všech uživatelů (Články) and Správa uživatelů tématu (Články). Below is a table for adding activities:

Články	Testy	Soubory	Diskuse
Vyberte...	Vyberte...	Vyberte...	Vyberte...
Vložit	Vložit	Vložit	Vložit
- Poslední lekce**: Opět si můžete vyzkoušet, co jste se dozvěděli a zadiskutovat si. Includes links for Uživatelé portálu (Testy) and Tak co vy na to (Diskuse).

Zobrazení článku

The screenshot shows a web browser window with the address bar displaying "Vyroubalův e-learning - Mozilla Firefox". The page has a blue header with navigation links: Soubor, Úpravy, Zobrazit, Přejít, Záložky, Nástroje, Nápořádá. Below the header is a large orange banner with the text "Mgr. MICHAL VYROUBAL" in white. To the right of the banner is a vertical navigation menu with links: Home, Lekce, Články, Testy, Soubory, Diskuse, and a button "Přihlásit se - Zaregistrovat se" with a timestamp "pátek, 24. únor 2006, 10:32:17".

The main content area is white and contains an article titled "Vyroubal: log-in-out a registrace". The article text is as follows:

Odhodíme anonymitu a vyměříme ji za registrovanou anonymitu.

Pro přihlášení a registraci slouží odkazy v pravém horním rohu. Jakmile jeden z těchto odkazů vyberete, portál je schopen přejít k šifrované komunikaci (SSL), takže vaše uživatelské jméno ani heslo nebude moci nikdo odposlechnout při jeho putování po drátě.

Je v tom samozřejmě jeden háček tato funkce je přístupná poze v případě, že portál je hostován na serveru, který mu SSL komunikaci umožní. Jak to poznáte? Podle protokolu uvedeného v adrese (URL).

http:// - nešifrované spojení

https:// - šifrované spojení

Jakmile se přihlásíte, je v pravém horním rohu odkaz pro odhlášení - ten Vás odhlásí a přepne do nešifrovaného spojení.

Rovněž se Vám zpřístupní odkaz pro změnu registračních údajů. K tomu jediná poznámka. Pokud nevyplníte pole pro heslo, zachová se Vám původní heslo.

Rada na závěr: Nezapomínejte heslo.

At the bottom of the article is a button "Zpět do kurzu" and a link "Seznam článků".

Below the article is a sidebar with the following sections:

- Skrýt seznam kurzů**
Vyroubalův elearning > Così jako návod > 1. portál > 2. uživatelé
- Role ve vybraném kurzu:
Čtenář (nepřihlášen)
- Přejít k nadřazenému kurzu**
- Seznam kurzů**
- 2. uživatelé**
- Zobrazit všechny kurzy

At the bottom of the page is a footer with the text "Adresa tohoto příspěvku: http://server6/diplomka/index.php?modul=3&otec=7&clanky_show=1&clanky_clanek=9" and "Copyright © 2005 Mgr. Michal Vyroubal. Všechna práva vyhrazena." To the right of the footer are logos for W3C XHTML 1.0 and W3C CSS.

Pokus nepřihlášeného uživatele o vstup do soukromého kurzu

The screenshot shows a web browser window with the address bar displaying "Vyrubalův e-learning - Mozilla Firefox". The page has a blue header with navigation links: Soubor, Úpravy, Zobrazit, Přejít, Zálžky, Nástroje, Nápořádá. The main content area is light blue and features the text "Mgr. MICHAL VYRUBAL" in large, bold, orange letters. Below this, there is a section titled "Přístup zamítnut" (Access denied) with the message: "Přihlašete se svým uživatelským jménem a heslem." (Log in with your username and password). Below this message are two orange buttons: "Přihlásit se" (Log in) and "Zaregistrovat se" (Register). A second message states: "Pokud nemáte uživatelské jméno, zaregistrujte se jako uživatelé portálu." (If you do not have a username, register as a portal user). At the bottom of the page, there is a footer with the text "Adresa tohoto příspěvku: http://server6/diplomka/index.php?modul=2&otec=15" and "Copyright © 2005 Mgr. Michal Vyrubal. Všechna práva vyhrazena." (All rights reserved). The footer also includes logos for W3C XHTML 1.0 and W3C CSS.

Mgr. MICHAL VYRUBAL

Přístup zamítnut

Přihlašete se svým uživatelským jménem a heslem.

Přihlásit se

Zaregistrovat se

Pokud nemáte uživatelské jméno, zaregistrujte se jako uživatelé portálu.

Skrýt seznam kurzů

Vyrubalův elearning >
Zátěžová herna > matematika

Role ve vybraném kurzu:
Bez přístupu (pouze pro přihlášené)

Přejít k nadřazenému kurzu

Seznam kurzů

matematika
rovnice
kvadratické rovnice
...
zlomky

Zobrazit všechny kurzy

Adresa tohoto příspěvku: <http://server6/diplomka/index.php?modul=2&otec=15>

Copyright © 2005 Mgr. Michal Vyrubal. Všechna práva vyhrazena.

W3C XHTML 1.0 W3C CSS

Pokus přihlášeného uživatele o vstup do soukromého kurzu

The screenshot shows the Vyroubalův e-learning portal in a Mozilla Firefox browser window. The page has a blue header with navigation links: Soubor, Úpravy, Zobrazit, Přejít, Zálžky, Nástroje, Nápořádá. The main content area is light blue with a large orange banner for 'Mgr. MICHAL VYROUBAL'. Below the banner, there are several sections:

- Skrýt seznam kurzů**: A list of courses including 'Vyroubalův elearning > Zátěžová herna > matematika'. Below this is a form to enter a role and a password, with a 'Zadejte vstupní klíč:' label and an 'Odeslat' button.
- Přijít k nadřazenému kurzu**: A section for accessing a parent course.
- Seznam kurzů**: A list of courses including 'matematika', 'rovnice', 'kvadratické rovnice', '...', and 'zlomky'.
- Zobrazit všechny kurzy**: A section for displaying all courses.

The right sidebar contains a navigation menu with links: Home, Lekce, Články, Testy, Soubory, Diskuse, and a login/logout link. Below the menu, there is a section titled 'Přístup zamítnut' (Access denied) with a message: 'Ke vstupu do tohoto tématu je vyžadována znalost hesla - vstupního klíče. Zadejte vstupní klíč k tomuto kurzu do políčka vlevo. Pokud vstupní klíč neznáte, dotazte se správců kurzu (viz níže).' Below this, there is a table showing the course details:

O tento kurz se starají:	
Vlastníci	Redaktoři
vyroubal (Michal Vyroubal)	Žádný redaktor
	Žádný příspěvatel

At the bottom of the page, there is a footer with the address 'Adresa tohoto příspěvku: https://server6/diplomka/index.php?modul=1&otec=15', copyright information 'Copyright © 2005 Mgr. Michal Vyroubal. Všechna práva vyhrazena.', and logos for W3C XHTML 1.0 and W3C CSS.

Automatické vyhodnocení testu

The screenshot shows a web browser window with the address bar displaying "Vyroubalův e-learning - Mozilla Firefox". The page has a blue header with navigation links: Soubor, Úpravy, Zobrazit, Přejít, Zálžky, Nástroje, Nápořádá. Below the header is a large orange banner with the text "Mgr. MICHAL VYROUBAL" and a navigation menu: Home, Lekce, Články, Testy, Soubory, Diskuse. The main content area is titled "Vyhodnocení testu" and displays the results of a test for the user "Mgr. MICHAL VYROUBAL". The test is titled "Test: Uživatelé portálu" and the user's score is "Váš test byl odeslán.".

Mgr. MICHAL VYROUBAL

Vyhodnocení testu
Váš test byl odeslán.

Test: Uživatelé portálu

Šifrované spojení	Požadovaný výsledek
<p>Portál je schopen pracovat se šifrovaným přenosem, do kterého se přepne při přihlašování uživatele. Tato technologie se jmenuje:</p> <p> <input type="checkbox"/> SSM <input type="checkbox"/> SRPŠ <input checked="" type="checkbox"/> SSL <input type="checkbox"/> UTP <input type="checkbox"/> TCP <input type="checkbox"/> IP </p>	<p>Požadovaný výsledek:</p> <p> <input type="checkbox"/> SSM <input type="checkbox"/> SRPŠ <input checked="" type="checkbox"/> SSL <input type="checkbox"/> UTP <input type="checkbox"/> TCP <input type="checkbox"/> IP </p> <p>Body: 1 (max: 1, penalizace: 0.1)</p>
<p>Blokace uživatele</p> <p>V době, kdy je uživatel blokovan, se může přihlásit, ale nebude mu dovoleno přispívat do diskuse.</p> <p>Pravda</p>	<p>Požadovaný výsledek:</p> <p>Špatně</p> <p>Body: -0.1 (max: 1, penalizace: 0.1)</p>
<p>Správa všech uživatelů</p> <p>Všechny uživatele portálu může spravovat uživatel, který má k hlavnímu tématu portálu právo... redaktor</p>	<p>Požadovaný výsledek:</p> <p>vlastník</p> <p>Nezáleží na velikosti písmen.</p> <p>Body: -0.1 (max: 1, penalizace: 0.1)</p>

Skrýt seznam kurzů
Vyroubalův e-learning > Cosi jako návod > 1. portál > 2. uživatelé

Role ve vybraném kurzu: Zděděné vlastnictví

Přejít k nadřazenému kurzu
Editace uživatelů tohoto kurzu
Přidat sem kurz
Upravit tento kurz
Smazat tento kurz

Seznam kurzů
2. uživatelé
Zobrazit všechny kurzy
Oblíbené kurzy
Přidat tento kurz k oblíbeným
Kurzy, ke kterým mám právo